Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT cheetah

Conference Paper · September 2017

DOI: 10.1109/IROS.2017.8206268

CITATIONS
20

READS
1,419

3 authors:

Gerardo Bledt
Massachusetts Institute of Technology
15 PUBLICATIONS **205** CITATIONS
SEE PROFILE

Patrick Wensing
University of Notre Dame
59 PUBLICATIONS **873** CITATIONS
SEE PROFILE

Sangbae Kim
Massachusetts Institute of Technology
78 PUBLICATIONS **5,872** CITATIONS
SEE PROFILE

Policy-Regularized Model Predictive Control to Stabilize Diverse Quadrupedal Gaits for the MIT Cheetah

Gerardo Bledt, Patrick M. Wensing, and Sangbae Kim

***Abstract*— This paper introduces a new policy-regularized model-predictive control (PR-MPC) approach to automatically generate and stabilize a diverse set of quadrupedal gaits. Model-predictive methods offer great promise to address balance in dynamic robots, yet require the solution of challenging non-linear optimization problems when applied to legged systems. The new proposed PR-MPC approach aims to improve the conditioning of these problems by adding regularization based on heuristic reference policies. With this approach, a unified MPC formulation is shown to generate and stabilize trotting, bounding, and galloping without retuning any cost-function parameters. Intuitively, the added regularization biases the solution of the MPC towards common heuristics from the literature that are based on simple physics. Simulation results show that PR-MPC improves the computation time and closed-loop outcomes of applying MPC to stabilize quadrupedal gaits.**

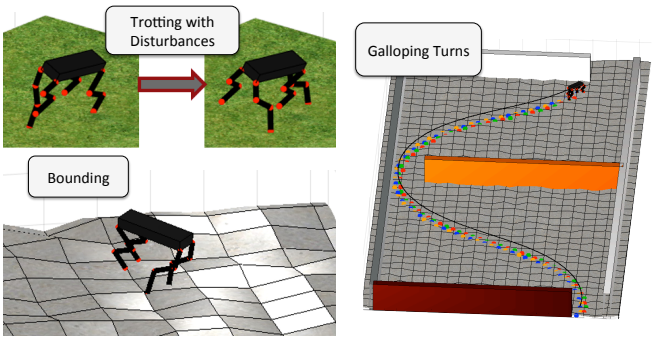

Fig. 1. This paper presents a predictive control framework for online control of dynamic 3D quadrupedal gaits. A single set of algorithm parameters stabilizes gaits from trotting to galloping when applied to a simplified Cheetah model.

I. Introduction

The challenge of dynamic locomotion is well captured by the fact that high-speed gaits are never instantaneously balanced in a traditional sense. Instead, dynamically stable systems are characterized by an ability to embrace their momentum while continually guiding the system into the future along a safe set of states. As a result, dynamic balance is inherently linked to accessible *future* states. Towards addressing this challenge, the development of predictive capabilities for legged robots may prove an important step in realizing a wider envelope of dynamic performance.

Prediction has been hypothesized to play an important role across sensorimotor control for humans and animals [1]. Internal models of musculoskeletal dynamics are hypothesized to exist in the brain for the support of predictive faculties. Despite the evidence in favor of this idea, results concerning our perception of geometry [2] cast doubt that such internal models adhere to the detailed physical modeling that often underlies predictive methods in robotics. Approximate internal models, akin to in [3], undoubtedly play a role in sensorimotor control.

Within robotics, an increasing focus has been placed on model-predictive control (MPC) methods in recent years. Modern approaches commonly provide model predictive control through solving linear, quadratic, or nonlinear optimization problems which determine control sequences over a receding prediction window into the future. Applications across hoppers [4], bipeds [5], humanoids [6], [7], and quadrupeds [8] alike have shown the power of predictive methods to plan and stabilize complex dynamic movements. Model predictive methods have also offered an ability to address many of the kinematic (joint limit, self-collision) and dynamic (torque limit, contact friction) constraints that challenge real-world robots.

Despite the success of these previous methods in robotics, the computational demands of MPC have limited its application in dynamic high-degree-of-freedom (DoF) systems. Within this domain, real-time constraints present a significant additional challenge. The use of simple models, or templates [9], has proven effective to plan the most salient aspects of system's dynamics and meet real-time constraints. Bipedal walking trajectory generation focused on the CoM has been treated with known footstep orientations using linear MPC [10] and with optimized orientations using non-linear MPC [11]. Simple descriptions for the CoM dynamics in running have enabled graceful real-time disturbance recovery [12], [13] in simulation experiments.

Beyond CoM templates, other approaches have attempted to incorporate angular aspects of the system evolution into simple models for planning and control. Templates which incorporate an angular state have been proposed [14], but their general application to control high-DoF robots remains open. In similar work, the control of centroidal angular momentum [15] has gained much attention as a heuristic for balance. The evolution of the centroidal momentum is linked to the net external wrench [16], which can be exploited to accelerate off-line whole-body trajectory optimization [17] or simplify linear-quadratic control of balance [18].

Previous work by the authors has pursued *on-line* methods to plan external forces for dynamic quadrupedal locomotion. Balancing the vertical impulses from contact and gravity enabled high-speed bounding across a range of speeds [19] with the MIT Cheetah. Results on autonomous jumping [20] used real-time optimization of forces with a planar rigid-

Authors are with the Department of Mechanical Engineering, MIT: {gbledt, pwensing, sangbae}@mit.edu

body model to generalize jumping for different obstacle heights. The use of simple models in these accomplishments provides strong evidence for the utility of a template-based control design philosophy. Although these results with the MIT Cheetah were untethered, underlying locomotion plans were developed on purely planar template models. Out-of-plane effects were handled purely through virtual-model feedback online. The work herein seeks to extend a template-based control design philosophy to 3D planning for diverse quadrupedal gaits.

The main contribution of this work is to describe a new approach which uses a reference policy to add regularization to model predictive control (MPC). More specifically, policy regularization is added to direct collocation optimization problems that are solved online for MPC. Generally, regularization is a process of imposing additional structure on the solution of an optimization problem with constraints or cost functions that penalize the solution complexity [21].

The proposed policy-regularized MPC (PR-MPC) imposes this structure by penalizing the deviation from a set of simple heuristic policies within direct collocation. Both the heuristic controls alone, and regularization-free MPC approaches have limitations that are addressed by this new approach. Heuristic policies are difficult to tune to a wide range of behaviors, and optimization problems for regularization-free MPC can suffer from long solve times and have many local optima. Intuitively, PR-MPC guides the MPC towards simple heuristic control choices that are common in the literature, while allowing the freedom to deviate when heuristics alone are not sufficient. Without retuning any cost functions, the approach is applicable for model predictive control of quadruped locomotion with a variety of 3D gaits.

The remainder of the paper is organized as follows. Section II presents a template model of the dynamics of the MIT Cheetah robot and describes a further-simplified model that is used in control design. Section III presents a formulation of nonlinear model predictive control for this simplified model which is able to stabilize a wide range of gaits in the template. A policy regularizer is included to guide the optimization towards heuristic reference policies through penalization. Section IV shows results of this approach, detailing its closed-loop performance and computational requirements for multiple gaits. A supplementary video is included to demonstrate the capabilities of the controller. Finally, Section V provides a short concluding discussion.

II. Template Model and Simplification

This section defines two simple models of the MIT Cheetah robot that capture its main dynamics. The first model, a template used for simulation, is a single rigid body in $\mathsf{SE}(3)$ that approximates the Cheetah as having massless legs. The dynamics of the model follow from straightforward first-principles physics. A second model, used in control design, provides an approximation to the dynamics of a single rigid body. This simplification helps to decrease the solve times of direct collocation optimization problems for MPC, while retaining the main dynamic effects important to locomotion.

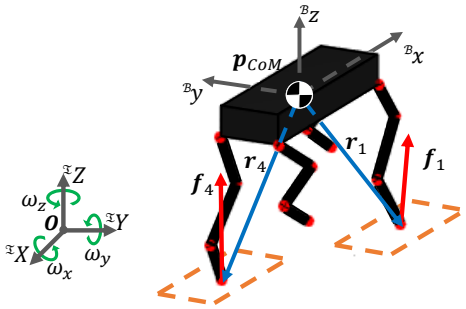

Fig. 2. Coordinate systems and definitions for the rigid body model in 3D. The vectors $\boldsymbol{r}_j$ specify the position of each foot relative to the body CoM, while forces $\boldsymbol{f}_j$ provide the force under each foot.

A. Simulation Model: Dynamics of the Rigid-Body in $\mathsf{SE}(3)$

The dynamics of the MIT Cheetah can be faithfully represented as a single rigid-body with fixed inertia. The legs of the robot have low mass and inertia compared with the main body, comprising only approximately 10% of the total mass. Thus, the state of the robot is defined through

$$\boldsymbol{x} := \begin{bmatrix} \boldsymbol{p}_c & {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}} & \dot{\boldsymbol{p}}_c & {}^{\mathcal{B}}\boldsymbol{\omega} \end{bmatrix} \tag{1}$$

where $\boldsymbol{p}_c \in \mathbb{R}^3$ provides the position of the body CoM, ${}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}} \in \mathsf{SO}(3)$ the orientation of the body frame $\mathcal{B}$ with respect to the inertial frame $\mathcal{I}$, and ${}^{\mathcal{B}}\boldsymbol{\omega} \in \mathbb{R}^3$ the angular velocity of the body. Throughout, leading superscripts specify the coordinate frame of reference. When omitted, the inertial frame $\mathcal{I}$ is assumed.

At each instant, the model is propelled by forces $\boldsymbol{f}_j \in \mathbb{R}^3$ acting at foot locations $\boldsymbol{r}_j \in \mathbb{R}^3$ relative to the CoM. Foot positions relative to the inertial frame origin are denoted as $\boldsymbol{p}_j = \boldsymbol{p}_c + \boldsymbol{r}_j$. Definitions are shown in Figure 2. Viewing both forces and relative footstep locations as important quantities for planning, these variables are collected in an input vector $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{f}_1 & \dots & \boldsymbol{r}_4 & \boldsymbol{f}_4 \end{bmatrix}$.

The net wrench created through the interaction of the feet with the ground includes terms which are bilinear in the forces $\boldsymbol{f}_j$ and contact locations $\boldsymbol{r}_j$. The net force and net moment about the CoM, $\boldsymbol{f}$ and ${}^{\mathcal{B}}\boldsymbol{\tau}$ respectively, are given as

$$\begin{Bmatrix} \boldsymbol{f} \\ {}^{\mathcal{B}}\boldsymbol{\tau} \end{Bmatrix} = h(\boldsymbol{u}) = \sum_{j=1}^{4} \begin{bmatrix} \mathbf{I} \\ {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^T \left[\boldsymbol{r}_j\right]_\times \end{bmatrix} \boldsymbol{f}_j \tag{2}$$

where $\left[\boldsymbol{r}_j\right]_\times \in \mathsf{so}(3)$ is the skew symmetric matrix such that $\left[\boldsymbol{r}_j\right]_\times \boldsymbol{b} = \boldsymbol{r}_j \times \boldsymbol{b}$, for any $\boldsymbol{b} \in \mathbb{R}^3$. The full dynamics of the rigid-body then follow common formula

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = \begin{Bmatrix} \dot{\boldsymbol{p}}_c \\ {}^{\mathcal{I}}\dot{\boldsymbol{R}}_{\mathcal{B}} \\ \ddot{\boldsymbol{p}}_c \\ {}^{\mathcal{B}}\dot{\boldsymbol{\omega}} \end{Bmatrix} = \begin{Bmatrix} \dot{\boldsymbol{p}}_c \\ {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}} \left[{}^{\mathcal{B}}\boldsymbol{\omega}\right]_\times \\ \frac{1}{m}\boldsymbol{f} - \boldsymbol{g} \\ {}^{\mathcal{B}}\bar{\boldsymbol{I}}^{-1} \left({}^{\mathcal{B}}\boldsymbol{\tau} - {}^{\mathcal{B}}\boldsymbol{\omega} \times {}^{\mathcal{B}}\bar{\boldsymbol{I}}\, {}^{\mathcal{B}}\boldsymbol{\omega} \right) \end{Bmatrix} \tag{3}$$

where $\boldsymbol{g} \in \mathbb{R}^3$ is the gravitational acceleration. The inertia tensor ${}^{\mathcal{B}}\bar{\boldsymbol{I}} \in \mathbb{R}^{3\times3}$ is fixed in body coordinates.

B. Control Design Model: Approximate Discrete Dynamics of a Rigid Body

The dynamics of a rigid body on $\mathsf{SE}(3)$ include significant nonlinearities to be handled in Model Predictive Control (MPC). While nonlinear MPC can be accomplished in principle, in practice, numerical solution approaches are often complicated by local minima and real-time constraints which challenge online application. Towards increasing the viability of online MPC, this section formulates simplified discrete equations of the previous model. These simplified dynamics are used to design controllers, while the full dynamics are used to evaluate controllers. Most importantly for control design, nonlinearities crucial to legged locomotion, from the feet interacting with the ground, are kept within the formulation. To signify that the simplified model uses approximations, all quantities will be denoted with hats $\hat{\cdot}$ above the associated variables.

The orientation of the robot is described in the control model through Euler angles as $\boldsymbol{\Theta} = [\theta, \phi, \psi]^T$, with an earth-fixed roll (θ), pitch (ϕ), yaw (ψ) convention. While we consider motions with non-zero pitch and roll, their effects on the *instantaneous* dynamics of the system can can be simplified. Consider the net external torque in body coordinates from (2).

$$^{\mathcal{B}}\hat{\boldsymbol{\tau}} = \sum_{j=1}^{4} {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{T} \left[\boldsymbol{r}_j\right]_{\times} \boldsymbol{f}_j$$

Under well-controlled locomotion, the pitch and roll of the base body are bounded above and below by a small value, meaning that the $^{\mathcal{B}}z$ axis of the robot's body frame shown in Figure 2 remains close to parallel with the $^{\mathcal{I}}Z$ axis of the inertial world frame. With this in mind, the transformation of the net moment into body coordinates can be approximated by noting that

$$\lim_{(\hat{\theta},\hat{\phi})\to(0,0)} {}^{\mathcal{I}}\hat{\boldsymbol{R}}_{\mathcal{B}}(\hat{\boldsymbol{\Theta}}) = \boldsymbol{R}_z(\hat{\psi}) \tag{4}$$

where $\boldsymbol{R}_z(\hat{\psi})$ denotes a rotation of $\hat{\psi}$ radians about the $^{\mathcal{I}}Z$ axis. This provides an approximated version of (2) as

$$\hat{h}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) := \begin{Bmatrix} \hat{\boldsymbol{f}} \\ {}^{\mathcal{B}}\hat{\boldsymbol{\tau}} \end{Bmatrix} = \sum_{j=1}^{4} \begin{bmatrix} \mathbf{I}_3 \\ \boldsymbol{R}_z^T(\psi) [\hat{\boldsymbol{r}}_j]_{\times} \end{bmatrix} \hat{\boldsymbol{f}}_j. \tag{5}$$

Again, such an approximation will only be valid provided that $^{\mathcal{B}}z$ remains within some reasonably bounded cone along the $^{\mathcal{I}}Z$ axis. This approximate body torque $^{\mathcal{B}}\hat{\boldsymbol{\tau}}$ provides an approximate angular acceleration

$$^{\mathcal{B}}\dot{\hat{\boldsymbol{\omega}}} = {}^{\mathcal{B}}\bar{\boldsymbol{I}}^{-1} {}^{\mathcal{B}}\hat{\boldsymbol{\tau}} \tag{6}$$

where velocity product terms $^{\mathcal{B}}\boldsymbol{\omega} \times {}^{\mathcal{B}}\bar{\boldsymbol{I}}\, {}^{\mathcal{B}}\boldsymbol{\omega}$ are neglected since they are small in practice.

The approximate alignment of the body frame and a yaw-rotated frame can also be used to simplify the dynamics of the Euler angle rates. The body angular velocity $^{\mathcal{B}}\boldsymbol{\omega}$ is related to Euler angle rates $\dot{\boldsymbol{\Theta}}$ through

$$\dot{\boldsymbol{\Theta}} = \boldsymbol{B}(\boldsymbol{\Theta})\, {}^{\mathcal{B}}\boldsymbol{\omega} \tag{7}$$

where $\boldsymbol{B}(\boldsymbol{\Theta}) \in \mathbb{R}^{3\times 3}$ is an Euler-angle rate matrix [22]. Although this matrix has a complex form in general, using the previous assumption, for states where roll and pitch are small, it can be approximated from

$$\lim_{(\theta,\phi)\to(0,0)} \boldsymbol{B}(\boldsymbol{\Theta}) = \mathbf{I}_3\,, \tag{8}$$

giving $\dot{\boldsymbol{\Theta}} \approx {}^{\mathcal{B}}\boldsymbol{\omega}$. Note that all of these approximations are applied to the dynamics of the system, allowing non-zero pitch and roll evolution over time, but neglecting their effects on the instantaneous dynamics. With these approximations, the transformation of torques to the body frame means that no nonlinearities in the approximate model dynamics appear outside of (5).

Remark 1. *It is important to note that these approximations to the dynamics are only included in the model of the robot that is used to generate controllers. When evaluating the controllers in simulation, the full dynamic effects from roll, pitch, and Coriolis forces are modeled. The admissibility of these approximations for control design is demonstrated in Section IV, showing applicability to design gaits such as bounding and galloping with non-zero pitch and roll. Intuitively, the approximations are admissible for control development, since the effects of pitch, roll, and body angular velocity on the dynamics $f(\cdot)$ in* (3) *are small in comparison to the effects of external forces on the system.*

Letting

$$\hat{\boldsymbol{x}} := \begin{bmatrix} \hat{\boldsymbol{p}}^T & \hat{\boldsymbol{\Theta}}^T & \dot{\hat{\boldsymbol{p}}}^T & {}^{\mathcal{B}}\dot{\hat{\boldsymbol{\omega}}}^T \end{bmatrix}^T$$

denote the approximate state, its dynamics are discretized as

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{A}_i \hat{\boldsymbol{x}}_i + \boldsymbol{B}_i \hat{h}\big(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i\big) + \boldsymbol{d}_i, \tag{9}$$

where $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are both fixed matrices and $\boldsymbol{d}_i$ a fixed vector dependent on the discrete time step, Δt_i

$$\boldsymbol{A}_i = \begin{bmatrix} \mathbf{I}_6 & \Delta t_i \mathbf{I}_6 \\ 0 & \mathbf{I}_6 \end{bmatrix}, \quad \boldsymbol{B}_i = \begin{bmatrix} \frac{\Delta t_i^2}{2}\, {}^{\mathcal{B}}\boldsymbol{I}^{-1} \\ \Delta t_i\, {}^{\mathcal{B}}\boldsymbol{I}^{-1} \end{bmatrix}, \quad \boldsymbol{d}_i = \begin{bmatrix} \frac{\Delta t_i^2}{2}\, \boldsymbol{a}_g \\ \Delta t_i\, \boldsymbol{a}_g \end{bmatrix} \tag{10}$$

with $^{\mathcal{B}}\boldsymbol{I} = \mathrm{diag}(m\mathbf{I}_3, {}^{\mathcal{B}}\bar{\boldsymbol{I}})$ and $\boldsymbol{a}_g = [\boldsymbol{g}^T, \mathbf{0}_{3\times 1}^T]^T$. These discrete dynamics have a simple form, and thus are a good candidate for MPC optimization. Their function evaluations will be rapid and state transition constraints possess a great deal of sparsity.

Remark 2. *Again, note that this discretization is applied as a model for control development. Once controllers are designed, they are evaluated in simulation with the continuous dynamics from Section II-A. The admissibility of this discretization for control design can only be determined conclusively from such an evaluation. Results in Section IV will show that the approximation is admissible.*

III. Policy-Regularized Model-Predictive Control: PR-MPC

This section describes the formulation and solution of a policy-regularized model predictive control problem using the simplified dynamic model from the previous section.

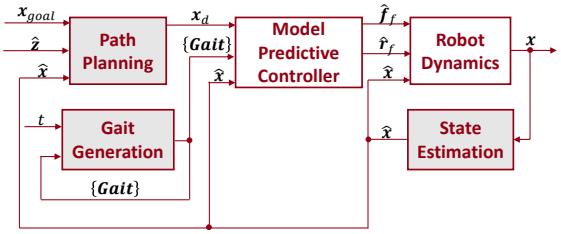

Fig. 3. Proposed Control Framework diagram for autonomous navigation with the MIT Cheetah. This paper addresses the design of the model predictive controller for selecting footstep locations and forces to follow a desired nominal trajectory using a fixed gait. Detailed description of the grey blocks in the system diagram represent components outside the scope of this work.

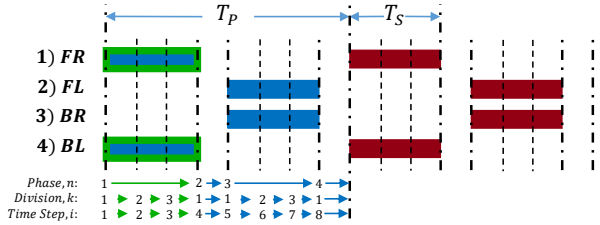

Fig. 4. Footfall pattern for a running trot gait showing the discretized divisions from setting the number of phases $N = 4$ and divisions per stance phase $K = 3$. The MPC optimizer plans forces for all N phases (green and blue). It executes the optimized plan only for the first phase (green), and then repeats the planning process. The stance time T_S and gait period T_P are used to construct heuristic control actions for policy regularization.

Figure 3 shows how this MPC is envisioned to ultimately fit in with an overall locomotion control architecture. A desired trajectory is assumed to be generated by a higher level path planning algorithm which addresses the combinatorial complexity of navigating safely in an environment with geometric obstacles. The goal of the MPC is to select footstep locations $\boldsymbol{r}_j$ and forces $\boldsymbol{f}_j$ which result in the robot accurately following this desired trajectory.

A. PR-MPC: A Direct Collocation Formulation

In order to find suitable controls, PR-MPC is formulated as a direct collocation optimization problem. Forward Euler integration constraints from (9) are used directly to place this trajectory optimization problem into a general form:

$$\min_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}} \sum_{i=0}^{T} \ell_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i) + \ell_T(\hat{\boldsymbol{x}}_T) \tag{11}$$

$$\text{subject to } \hat{\boldsymbol{x}}_{i+1} = \boldsymbol{A}_i\hat{\boldsymbol{x}}_i + \boldsymbol{B}_i\hat{h}(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i) + \boldsymbol{d}_i \tag{12}$$

$$\boldsymbol{c}_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i) \leq 0 \tag{13}$$

$$\boldsymbol{c}_{i'}(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_{i+1}, \hat{\boldsymbol{u}}_i, \hat{\boldsymbol{u}}_{i+1}) \leq 0 \tag{14}$$

where each ℓ_i provides the incremental cost for step i, ℓ_T the terminal cost, $\boldsymbol{c}_i$ encodes state and control constraints, and $\boldsymbol{c}_{i'}$ encodes transition constraints.[1]

The remainder of this section details the cost and constraints for PR-MPC. To change constraints based on gait, a fixed-timing gait description is passed to MPC as described in Section III-B. Section III-C describes cost functions and policy regularization, with heuristic reference policies detailed in Section III-D. Section III-E describes constraints.

B. Gait Description

The footstep gait pattern is nominally specified as a set of time-based phases. Each phase begins and ends at a scheduled event in the footstep planner where a foot touchdown or liftoff is desired to occur. Instead of using a fixed time prediction horizon, a number of future gait phases, N, is used to define the horizon. Within each phase, the time period is further discretized into a number of divisions per phase, K. The model (9) is purely linear during flight phases since there can be no ground reaction force when no feet touch the ground. Thus, as a special case, only 1 division is needed in a flight phase. This observation further reduces the number of variables necessary for direct collocation.

As an example, a footfall gait pattern for a running trot is shown in Figure 4 with a $N = 4$ phase prediction horizon and $K = 3$ divisions per stance phase. Although 4 phases are predicted with the model dynamics, only the optimal input for the first predicted phase is executed in simulation, as signified by the green in the figure. Once the optimal foot locations and ground reaction force profiles are carried out for the first phase, the controller re-predicts a new set of optimal controls for the next prediction horizon.

Remark 3. *Although this work considers fixed gait timing, recent methods have begun to address gait timing optimization in MPC through gradient [23] and non-gradient approaches [24]. These higher-level timing optimizers could be considered for cases where fixed timing is not sufficient.*

C. Incremental Cost - Incorporation of Policy Regularization

The incremental cost $\ell_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i)$ in (11) is designed to include policy regularization which guides the MPC to roughly follow simple policies from common heuristics in the literature. Given the desired state $\boldsymbol{x}_{d,i}$ and a reference control $\boldsymbol{u}_{ref,i}$ from heuristics, the incremental cost at time step i is designed as

$$\ell_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i) = \tilde{\boldsymbol{x}}_i^T\boldsymbol{Q}_i\tilde{\boldsymbol{x}}_i + \tilde{\boldsymbol{u}}_i^T\boldsymbol{R}_i\tilde{\boldsymbol{u}}_i, \text{ , where} \tag{15}$$

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_{d,i} - \hat{\boldsymbol{x}}_i \tag{16}$$

$$\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_{ref,i} - \hat{\boldsymbol{u}}_i \tag{17}$$

and terminal cost similarly as $\ell_T(\hat{\boldsymbol{x}}_T) = \tilde{\boldsymbol{x}}_T^T\boldsymbol{Q}_T\tilde{\boldsymbol{x}}_T$. The desired states $\boldsymbol{x}_{d,i}$ include both the desired body path, as well as zero desired roll and pitch. Thus, the first cost term $\tilde{\boldsymbol{x}}_i^T\boldsymbol{Q}_i\tilde{\boldsymbol{x}}_i$ roughly penalizes a lack of balance. The second term in the cost $\tilde{\boldsymbol{u}}_i^T\boldsymbol{R}_i\tilde{\boldsymbol{u}}_i$ is the policy regularization term. This second term penalizes the complexity of the solution in terms of its deviation of simple policies.

D. Reference Controls for Policy Regularization

A reference foot position for each step is designed based on a Raibert heuristic [25] with a capture-point-based [26]

[1]Equality constraints can be captured through this formulation without loss of generality by considering a pair of constraints functions which differ only in their sign.

feedback term. Nominally this footstep would be designed to occur at a point on the ground

$$\boldsymbol{p}_{ref,j} = \boldsymbol{P}^T \boldsymbol{P} \left(\boldsymbol{p}_{c,0} + \boldsymbol{p}_{h,j} + \frac{T_s}{2} \dot{\boldsymbol{p}}_{c,d} + \sqrt{\frac{z_0}{\|\boldsymbol{g}\|}} (\dot{\boldsymbol{p}}_c - \dot{\boldsymbol{p}}_{c,d}) \right) \quad (18)$$

where z_0 is a nominal height of locomotion, $\boldsymbol{P} = \left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right)$ is a projection matrix, $\boldsymbol{p}_{h,j}$ provides the position of hip j relative to the CoM, and $\boldsymbol{p}_{c,0}$ is the initial position of the CoM at the start of the step. For the purpose of constructing a reference control, the CoM is approximated to follow

$$\boldsymbol{p}_{c,k} = \boldsymbol{p}_{c,0} + \left(\tfrac{k-1}{K_n}\right) T_s \dot{\boldsymbol{p}}_{c,d} \quad (19)$$

at division k. This provides a reference vector of foot j relative to the CoM at division k as

$$\boldsymbol{r}_{ref,j,k} = \boldsymbol{p}_{ref,j} - \boldsymbol{p}_{c,0} - \left(\tfrac{k-1}{K_n}\right) T_s \dot{\boldsymbol{p}}_{c,d} \,. \quad (20)$$

Reference forces are constructed identically on each foot using the principle of vertical impulse scaling [19] and a conical pendulum model for centripetal force during turning [27]. These are given by

$$\boldsymbol{f}_{ref} = \frac{m T_P}{4 T_S} \begin{bmatrix} \cos(\psi_d)\, \dot{\psi}_d \, \|\dot{\boldsymbol{p}}_{c,d}\| \\ \sin(\psi_d)\, \dot{\psi}_d \, \|\dot{\boldsymbol{p}}_{c,d}\| \\ \|\boldsymbol{g}\| \end{bmatrix} \quad (21)$$

where T_P is the overall gait period, and T_S is the stance time for each foot. The first two terms encode centripetal force, the last encodes vertical force, and the leading coefficient ensures that the net vertical and lateral impulse are consistent with the physics of the desired motion.

E. State, Control, and Transition Constraints

To ensure feasibility of the generated trajectories, the MPC includes a number of state/control contraints, $\boldsymbol{c}_i$, defined in (13) and transition constraints, $\boldsymbol{c}_{i'}$ defined in (14). Constraints on each footstep are included to ensure that the foot is placed on the ground, which is assumed to occur at a fixed height $z = 0$ during MPC. Thus, the constraint takes the form:

$$^{\mathcal{I}}Z^T (\hat{\boldsymbol{p}}_c + \hat{\boldsymbol{r}}_j) = 0 \quad (22)$$

where $^{\mathcal{I}}Z$ is the unit vector shown in Figure 2. Kinematic reachability constraints are added to ensure that the foothold remains within the workspace throughout the gait

$$\|\boldsymbol{r}_{j,k} - \boldsymbol{p}_{h,j}\| \leq l_{max} \,. \quad (23)$$

Friction constraints are imposed with a linearized friction pyramid $\mathcal{P}_\mu$ using $\mu = 0.3$

$$\mathcal{P}_\mu = \{(f_x, f_y, f_z) \text{ s.t. } |f_x| \leq \mu f_z \text{ and } |f_y| \leq \mu f_z\}. \quad (24)$$

Transition constraints are added between divisions of the same stance phase to ensure that each foot position remains fixed. These take the form

$$\hat{\boldsymbol{p}}_{c,k+1} + \hat{\boldsymbol{r}}_{j,k+1} = \hat{\boldsymbol{p}}_{c,k} + \hat{\boldsymbol{r}}_{j,k} \quad (25)$$

across each division of continuous support for foot j, as specified by the gait description.

F. Direct Collocation Formulation: Justification

While the optimization problem (11) uses a direct collocation formulation and requires optimization over states and controls, PR-MPC could be implemented with other trajectory optimization solution approaches. However, the choice of direct collocation here is motivated by the formulation of many of the constraints described previously. By using a direct collocation formulation and defining the constraints to be dependent solely on adjacent time steps, as in (14), banded diagonal sparsity patterns within the cost and constraint Jacobians/Hessians can be exploited by commercial optimizers [28]. These computational benefits would cease to exist if state trajectories were not optimized along with controls, as would be the case for single shooting.

Beyond sparsity considerations, multiple shooting and direct collocation have marked benefits to solve trajectory optimization problems. As the solutions to nonlinear differential evolutions are often sensitive to small perturbations, single shooting trajectory optimization often suffers from a high degree of nonlinearity [28], requiring prohibitively small step sizes which challenge numerical convergence. Multiple shooting and direct collocation formulations take highly nonlinear constraints on the dynamic evolution of the system and make a trajectory optimization problem (roughly) "less nonlinear" through breaking dynamics constraints into smaller, less sensitive, windows of integration [28].

IV. Results

This section evaluates the PR-MPC approach through simulation experiments. Results are first given for PR-MPC applied to a diverse set of gaits. Following this verification, the benefits of PR-MPC are illustrated through comparative simulation without policy regularization. Finally, results are provided that demonstrate the applicability of policy regularization yet further for non-steady state gaits and turning.

A. PR-MPC for Diverse Gaits

The proposed PR-MPC was used to generate and stabilize bounding, trotting, and galloping in simulation. A single set of algorithm cost function parameters was used across these gaits, as detailed in Table I. Bounding and trotting used a prediction horizon of $N = 2$ phases with $K = 4$ discrete divisions per step. Galloping used a prediction horizon of $N = 3$ phases and $K = 4$ discrete divisions per phase. The video attachment shows the resulting gaits in simulation. The simulation was carried out as follows. At the beginning of each gait phase, the direct collocation PR-MPC optimization problem (11) was solved. Then, piecewise-continuous optimal controls $\mathbf{u}^*$ were applied to the continuous model from Section II through the use of `ode45` in MATLAB, with continuous simulation running through the remainder of the phase before PR-MPC optimization was re-run. Table II provides the system parameters used in the simulation.

Figure 5 shows how the PR-MPC is able to automatically adapt its footstep locations to the target gait. The controller adjusts fore-aft and lateral footsteps to minimize postural disturbances from body torques due to asymmetric stance foot

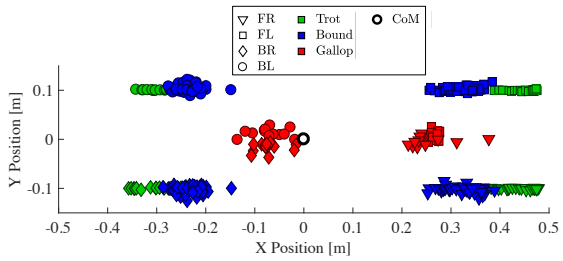

Fig. 5. Footfall placements (shown relative to the CoM) are generated automatically and specialized to the target gait sent to the MPC controller.

TABLE I

$\boldsymbol{Q}$ AND $\boldsymbol{R}$ WEIGHTING MATRIX VALUES

Parameter	Weight Vector			Units
$Q_{\hat{\boldsymbol{p}}}$	0	0	5000	$\left(\frac{1}{m}\right)^2$
$Q_{\hat{\boldsymbol{\Theta}}}$	500	1000	2500	$\left(\frac{1}{rad}\right)^2$
$Q_{\dot{\hat{\boldsymbol{p}}}}$	2500	2500	2500	$\left(\frac{s}{m}\right)^2$
$Q_{\mathcal{B}\dot{\hat{\boldsymbol{\omega}}}}$	2.5	2.5	300	$\left(\frac{s}{rad}\right)^2$
$R_{\hat{\boldsymbol{r}}}$	350	350	350	$\left(\frac{1}{m}\right)^2$
$R_{\hat{\boldsymbol{f}}}$	0.25	0.25	0.001	$\left(\frac{1}{N}\right)^2$

locations relative to the CoM. Galloping gaits automatically use footfalls close to the center of the body, while resulting trotting gaits observe placements of the feet directly in front of the hip on average. Across each of these gaits, the same reference policies were use to add policy regularization to the MPC problem. In the case of galloping, these heuristics are not as directly applicable due to the roll, pitch, and yaw induced by asymmetric footfall timing. Importantly, PR-MPC provides the flexibility to deviate from these heuristics where there is benefit to the stabilization of the body posture.

B. Comparative Benefits of PR-MPC

PR-MPC offers a number of benefits over purely applying heuristic controllers or applying MPC without policy regularization. From a high-level standpoint, heuristic controllers require a great deal of tuning, and often require tuning that is specific to each gait and/or operating point. The heuristics used here, in fact, are not enough to stabilize locomotion on their own. For instance, the reference force in (21), while based on first principles, is not enough to initiate changes in speed. However, these references are still able to be used to improve the timing and closed-loop outcomes in PR-MPC.

Table III describes a series of experiments to illustrate the merits of PR-MPC. In three cases, Cases A-C in the table, no policy regularization is used. Instead, heuristic controllers are applied to generate reference forces $\boldsymbol{f}_{ref}$ and foot positions $\boldsymbol{r}_{ref}$ which are used as initial seeds to the optimization. In the last case, case D, policy regularization is added to the MPC problem. The video attachment shows results for cases B, C, and D. In case A, the MPC was verified to be stuck in a local optima, and the resulting closed-loop control resulted in a fall. Footfall patterns for the remaining stable cases are shown in Fig. 6. Note that case B and case C have the same cost function, but different initial seeds. By changing the initial seed for the foot positions, these two cases exhibit convergence to different local optima. While case C is nominally well behaved, spurious footstep locations are observed. In comparison, PR-MPC demonstrates more predictable footstep patterns.

TABLE II

SYSTEM PARAMETERS OF THE SIMPLIFIED CHEETAH MODEL

Parameter	Value	Units
m	33.0	kg
I_{xx}	0.50	kg m^2
I_{yy}	2.90	kg m^2
I_{zz}	2.90	kg m^2
Body Width	0.20	m
Body Length	0.70	m
l_{max}	0.60	m

TABLE III

SIMULATION CASES

Case	$\hat{\boldsymbol{u}}_{\boldsymbol{f},seed}$	$\hat{\boldsymbol{u}}_{\boldsymbol{r},seed}$	$\boldsymbol{R}$
A	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
B	$\boldsymbol{f}_{ref}$	$\mathbf{0}$	$\mathbf{0}$
C	$\boldsymbol{f}_{ref}$	$\boldsymbol{r}_{ref}$	$\mathbf{0}$
D	$\boldsymbol{f}_{ref}$	$\boldsymbol{r}_{ref}$	$\boldsymbol{R}_{\hat{\boldsymbol{f}},\hat{\boldsymbol{r}}}$

PR-MPC also provides benefit to the solution time of the MPC problem. Intuitively, the addition of policy regularization improves the conditioning the direct collocation optimization problem. While further theoretical analysis is necessary, it is hypothesized that policy regularization decreases the sensitivity of the problem by biasing the solution towards known reasonable policies. Figure 7 shows the solution time for the different cases. PR-MPC is found to have the lowest solve time across each of the gaits considered, and further exhibits the smallest standard deviation in solve times. The solve timing numbers represent solution of (11) with `fmincon` and all function evaluations done in MATLAB. Current engineering efforts are underway to further specify sparsity patterns for the solver and to convert the solution to C/C++. These efforts are expected to decrease solve times by a factor of 10 based on past experience [20].

Note that as additional information (via initial seeds or policy regularization) is provided for the solution of the MPC problem, in general the solve times decrease. A notable exception is in going from case B to case C for galloping. In case C, a nonzero reference position is given as an initial seed for the foot touchdown locations. However, as shown in Fig. 5, the optimal foot touchdown locations for galloping deviate from the heuristic locations. Despite this disparity, policy regularization still improves the solve time of the galloping MPC.

C. Non-Steady-State Gaits

PR-MPC also has applicability to control non-steady-state gaits and more aggressive behaviors in 3D. To demonstrate the ability to handle disturbances, a push with net impulse $[-23.1, 8.25, 0]^T$ Ns was applied at the beginning of a trotting gait phase after the feet had been placed on the ground. The controller completed one step before the MPC was

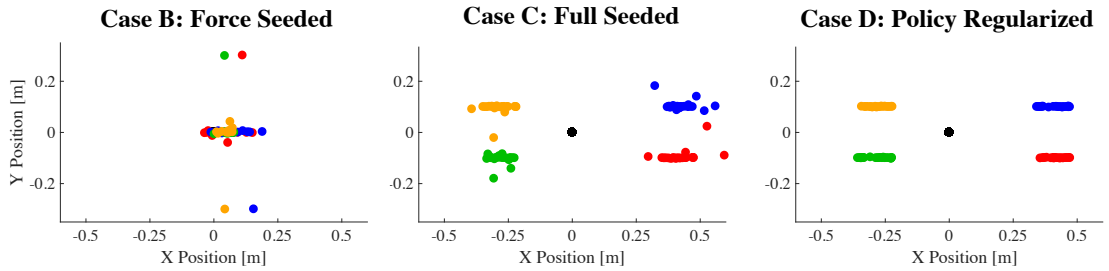

Fig. 6. Footstep locations resulting from solving MPC problems with different initial seeds and policy regularization. Further details on the different cases are provided in Table III.

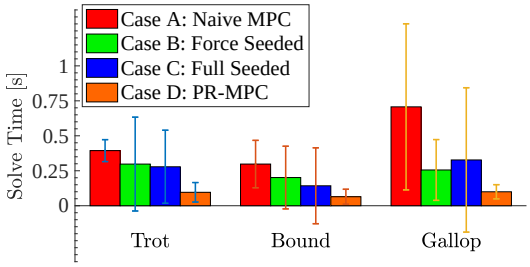

Fig. 7. Average solve time and standard deviation of direct collocation MPC problems with different initial seeds and policy regularization. Further details on the different cases are provided in Table III.

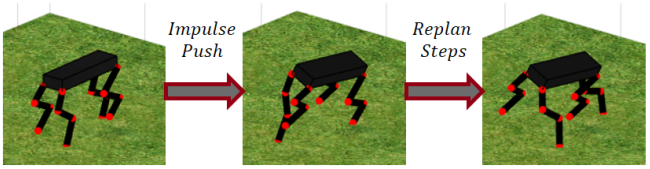

Fig. 8. Footstep planning with MPC allows the robot to choose new footstep locations autonomously to recover from disturbances.

used to select a new foot location and associated force profile for recovery. The results of this process is shown in Figure 8. Figure 9 shows state trajectories following this push disturbance, exhibiting large roll and pitch. Through these excursions, the net body torque approximation in (5) causes deviation between the control model and simulation. Despite these errors in the control model, stabilizing foot placements and forces are selected for closed-loop control. These results empirically support the admissibility of the approximations from Section II-B to generate appropriate controls.

A second test was preformed to show the ability of the controller to change speeds. From rest, the robot was given an accelerating desired forward speed up to $1\,^{\mathrm{m}}/_{\mathrm{s}}$ and a constant $0\,^{\mathrm{rad}}/_{\mathrm{s}}$ desired turn rate. Results of the forward speed tracking are shown in Figure 10. At 10 s, when the desired velocity becomes fixed, the controller converges to a slightly pitched downward nominal gait with an average pitch of approximately $5°$. There is a maximum roll angle of nominally $1°$ during the transient. Note that again, no cost function parameters where changed in comparison to the previous results, and the same heuristics were used for policy regularization.

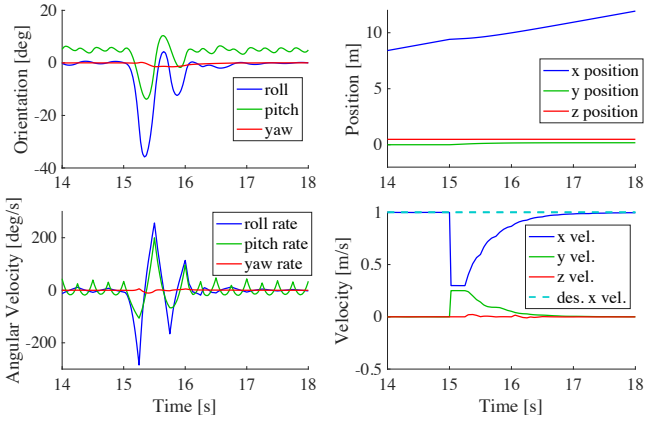

Fig. 9. State trajectories following a push disturbance at 15 s.

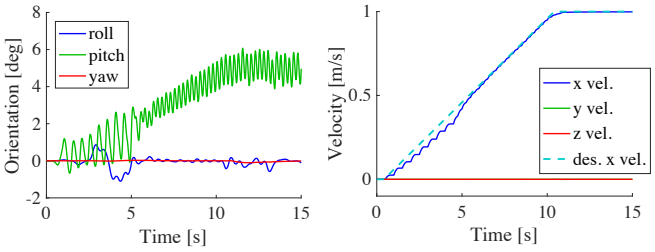

Fig. 10. State trajectories for an acceleration up to 1 m/s trotting.

The policy regularization approach is also applicable to generate and stabilize turning gaits through the conical pendulum heuristic (21). Fig. 11 shows state trajectories and tracking a trajectory with turning to the left and right during bounding. The PR-MPC results in emergent body roll into the turn. These collective results, across steady state gaits, disturbances, and transients support the generality of this PR-MPC to generate and stabilize quadrupedal locomotion.

V. Conclusion

This work has presented a new framework for policy-regularized model predictive control (PR-MPC) of legged robots. A strategic regularizer, based on heuristic reference policies from the literature was added to the cost of the trajectory optimizer to guide the solver towards controllers based in simple physics. With this approach, PR-MPC is able to stabilize a wide variety of gaits without retuning any algorithm parameters. Although the application of nonlinear MPC presents the need to solve challenging non-convex

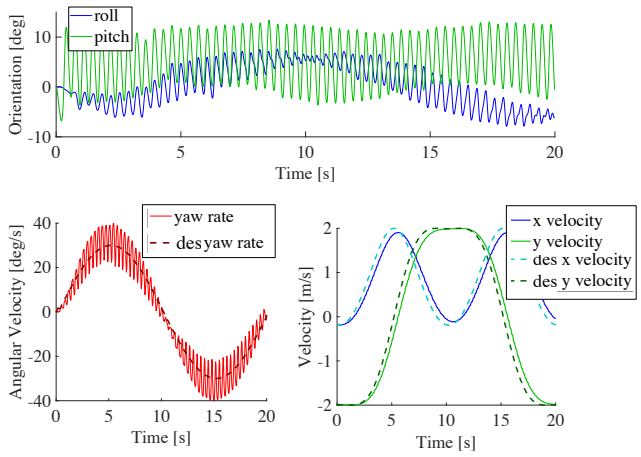

Fig. 11. Tracking a desired turning trajectory for 2 m/s bounding.

optimization problems, PR-MPC is an effective method to decrease the solve times of these methods and to decrease sensitivity to local optima. In a sense, PR-MPC borrows the best features from heuristic and MPC control, making use of heuristic knowledge but allowing optimization to tailor these heuristics to the specific details of each gait.

Future work will focus on the implementation of the proposed framework in a compiled language, ultimately leading to experiments with the MIT Cheetah robot. Future work will also explore the integration and implications of complementary constraints [29] within the formulation in order to enable emergent, rather than fixed, footfall sequences and timing. The proposed approach is envisioned to play a central role in expanding the adaptability of the Cheetah in uncertain and geometrically complex environments.

ACKNOWLEDGMENT

Funding for this work was supported in part through NSF award IIS-1350879 and by the Agency for Defense Development of Korea under contract UD140073ID.

REFERENCES

[1] M. Kawato, "Internal models for motor control and trajectory planning," *Current Opinion in Neurobiology*, vol. 9, no. 6, pp. 718 – 727, 1999.

[2] E. Fasse, N. Hogan, B. Kay, and F. Mussa-Ivaldi, "Haptic interaction with virtual objects: Spatial perception and motor control," *Biological Cybernetics*, vol. 82, no. 1, pp. 69–83, 2000.

[3] B. W. Andrews, E. D. Sontag, and P. A. Iglesias, "An approximate internal model principle: Applications to nonlinear models of biological systems," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 15873 – 15878, 2008. 17th {IFAC} World Congress.

[4] M. Rutschmann, B. Satzinger, M. Byl, and K. Byl, "Nonlinear model predictive control for rough-terrain robot hopping," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1859–1864, Oct. 2012.

[5] M. J. Powell, E. A. Cousineau, and A. D. Ames, "Model predictive control of underactuated bipedal robotic walking," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 5121–5126, May 2015.

[6] Y. Tassa, N. Mansard, and T. E., "Control-limited differential dynamic programming," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1168–1175, 2014.

[7] B. Henze, C. Ott, and M. A. Roa, "Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3253–3258, Sept 2014.

[8] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds," *ArXiv e-prints*, July 2016.

[9] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *J. Exp. Biol.*, vol. 202, no. 23, pp. 3325–3332, 1999.

[10] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Foot Step Placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[11] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Soures, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robotics and Automation Letters*, vol. 2, pp. 10–17, Jan 2017.

[12] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3D-SLIP model," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Rob. and Sys.*, (Tokyo, Japan), pp. 5134–5140, Nov. 2013.

[13] J. Englsberger, P. Kozlowski, and C. Ott, "Biologically inspired deadbeat control for running on 3d stepping stones," in *IEEE-RAS Int. Conf. on Humanoid Robots*, pp. 1067–1074, Nov 2015.

[14] S.-H. Lee and A. Goswami, "Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots," in *IEEE Int. Conf. on Rob. and Automation*, pp. 4667–4672, April 2007.

[15] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2, pp. 161–176, 2013.

[16] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application in dynamic whole-body control," *International Journal of Humanoid Robotics (Special Issue on Whole-Body Control)*, vol. 13, pp. 1550039:1–23, 2015.

[17] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 295–302, 2014.

[18] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *IEEE-RAS Int. Conf. on Humanoid Robots*, pp. 874–880, Nov 2015.

[19] H. W. Park, S. Park, and S. Kim, "Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3d running of mit cheetah 2," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5163–5170, May 2015.

[20] H.-W. Park, P. Wensing, and S. Kim, "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds," in *Proceedings of Robotics: Science and Systems*, (Rome, Italy), July 2015.

[21] M. J. Wainwright, "Structured regularizers for high-dimensional problems: Statistical and computational issues," *Annual Review of Statistics and Its Application*, vol. 1, pp. 233–253, 2014.

[22] K. Waldron and J. Schmiedeler, "Chapter 1: Kinematics," in *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), New York: Springer, 2008.

[23] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *ICRA*, 2017.

[24] C. Mastalli, M. Focchi, I. Havoutis, S. Calinon, J. Buchli, D. G. Caldwell, A. Radulescu, and C. Semini, "Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion," in *ICRA 2017*, 2017.

[25] M. H. Raibert, *Legged robots that balance*. Cambridge, MA, USA: MIT Press, 1986.

[26] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE-RAS Int. Conf. on Humanoid Robots*, (Genova, Italy), pp. 200–207, Dec. 2006.

[27] D. Krasny and D. Orin, "Evolution of a 3D gallop in a quadrupedal model with biological characteristics," *Journal of Intelligent & Robotic Systems*, vol. 60, pp. 59–82, 2010.

[28] M. Diehl, H. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions in Biomechanics and Robotics* (M. Diehl and K. Mombaur, eds.), vol. 340 of *Lecture Notes in Control and Information Sciences*, pp. 65–93, Springer Berlin Heidelberg, 2006.

[29] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.