Robust Predictive Control for Quadrupedal Locomotion: Learning to Close the Gap between Reduced- and Full-Order Models

Abhishek Pandala[1], Randall T. Fawcett[1], Ugo Rosolia[2], Aaron D. Ames[2], and Kaveh Akbari Hamed[1]

***Abstract*—Template-based reduced-order models have provided a popular methodology for real-time trajectory planning of dynamic quadrupedal locomotion. However, the abstraction and unmodeled dynamics in template models significantly increase the gap between reduced- and full-order models. This letter presents a computationally tractable robust model predictive control (RMPC) formulation, based on convex quadratic programs (QP), to bridge this gap. The RMPC framework considers the single rigid body model subject to a set of unmodeled dynamics and plans for the optimal reduced-order trajectory and ground reaction forces (GRFs). The generated optimal GRFs of the high-level RMPC are then mapped to the full-order model using a low-level nonlinear controller based on virtual constraints and QP. The proposed hierarchical control framework is employed for locomotion over rough terrains. We leverage deep reinforcement learning to train a neural network to compute the set of unmodeled dynamics for the RMPC framework. The proposed controller is finally validated via extensive numerical simulations and experiments for robust and blind locomotion of the A1 quadrupedal robot on different terrains.**

***Index Terms*—Legged Robots, Motion Control, Multi-Contact Whole-Body Motion Planning and Control**

I. Introduction

REAL-time trajectory planning and control algorithms have been popular methods to realize versatile and dynamic motions in quadrupedal robots rivaling their biological counterparts. Approaches along these lines can be sectioned into two categories: the ones using full-order models and the others using reduced-order (i.e., template-based) models. Reduced-order models provide a low-dimensional realization of the nonlinear full-order dynamics [1]. This makes the planning algorithms computationally tractable and amenable for real-time implementation. Furthermore, if the template models are linear or linearized, the planning algorithms can be transcribed as convex optimization problems. Planning dynamic locomotion that accommodates the full-order models suffers from a computational bottleneck arising from high-dimensionality. Owing to the low-dimensional approximation of the full-order dynamics, reduced-order models inherently ignore some of the rich full-order and nonlinear dynamics that may limit the full potential of the robot. This motivates the development of a planning and control framework that bridges the gap between reduced- and full-order models.

Towards this goal, this paper presents a layered control approach that learns to close the gap (see Fig. 1). At the higher layer, we propose a computationally tractable robust model predictive control (RMPC) framework that generates optimal reduced-order trajectories subject to a set of modeling uncertainties. This set of uncertainties may arise from abstraction and unmodeled dynamics, and is used to bridge the gap between reduced and full-order models. At the lower layer, a nonlinear controller, based on virtual constraints and quadratic programming (QP), is then developed to map the optimal reduced-order trajectories to the full-order locomotion model. The proposed layered control architecture can be integrated with deep reinforcement learning (RL) techniques to train a neural network (NN) that computes the set of uncertainties (a.k.a. the gap) for the RMPC algorithm. The robust control framework is finally validated via an extensive collection of numerical and experimental studies for the blind locomotion of the A1 quadrupedal robot on different terrains.

A. Background, Motivation, and Related Work

Reduced-order models can be easily integrated with the model predictive control (MPC) framework leading to more dynamic behaviors. Some of the popular reduced-order models include the linear inverted pendulum (LIP) model [2], centroidal dynamics [3], and single rigid body (SRB) dynamics [4]–[7] that have been employed for the real-time trajectory planning and control of various bipedal [8]–[11] and quadrupedal robots [4]–[6], [12]. Reduced-order models come with their own set of challenges. First, they ignore some of the full-order dynamics. For instance, the SRB model ignores the dynamics of legs. This is a valid assumption when the mass of the legs is insignificant compared to the total mass of the robot. The SRB dynamics have been integrated with the kinematics of the legs, but not the full-order dynamics in [13]–[15]. Second, when employing a reduced-order model on a legged robot, the optimal reduced-order trajectories have to be translated into full-order joint position or torque commands. In this regard, researchers have employed various techniques, most of them involving a hierarchical control structure with a

Manuscript received: January, 11, 2022; Revised April, 12, 2022; Accepted May, 6, 2022. This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments. The work of A. Pandala and K. Akbari Hamed is supported by the National Science Foundation (NSF) under Grant 1923216. The work of R. T. Fawcett is supported by the NSF under Grant 2128948. The work of A. D. Ames is supported by the NSF under Grant 1923239.

[1]A. Pandala, R. T. Fawcett, and K. Akbari Hamed (Corresponding Author) are with the Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA, 24061, USA, {agp19, randallf, kavehakbarihamed}@vt.edu

[2]U. Rosolia and A. D. Ames are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125, USA, ugo.rosolia@gmail.com, ames@caltech.edu

Digital Object Identifier (DOI): see top of this page.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3176105, IEEE Robotics and Automation Letters

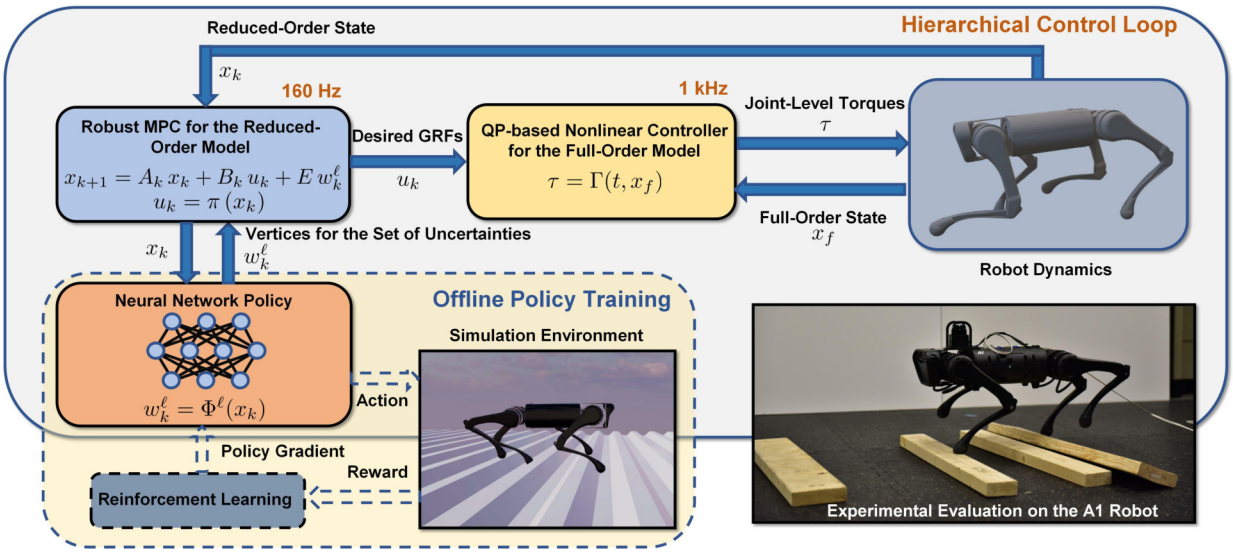

Fig. 1. Overview of the proposed hierarchical control algorithm, based on RMPC, NN, and QP-based nonlinear controller. The real-time control loops and signals are shown by the solid blocks and arrows, respectively. The offline training procedure is shown by the dashed blocks and signals.

high-level MPC and a low-level controller. For example, [4], [6], [7] have used Jacobian mapping, [16], [17] have used hybrid zero dynamics (HZD)-based controllers, [18] has used control barrier functions, and [13], [15] have used joint space whole-body controllers.

The problem of addressing uncertainties in legged locomotion has been predominantly approached at the low level. These approaches rely on the inherent robustness of the template dynamics integrated with a nominal MPC. Uncertainties are then addressed by making modifications to the existing low-level controller. For instance, the problem of addressing external disturbances such as forces is done by estimating them with conjugate momentum-based force observers [19], [20]. Input to state stability (ISS) has also been used to quantify uncertainty in walking robots [21]. Few other approaches have made use of frequency shaping [14], [22] to penalize high-frequency terms in the cost function to address locomotion on soft ground. However, these approaches do not directly accommodate disturbances at the high level.

In the MPC community, various popular approaches exist that consider uncertainties and robustness including closed-loop (feedback) min-max MPC [23], [24], open-loop min-max MPC [25], [26], and tube MPC [27], [28]. Open-loop MPC approaches determine control actions at every time sample by optimizing a performance criterion and addressing uncertainty in both cost minimization and constraint satisfaction. This leads to a small domain of feasibility and conservative control actions. Feedback MPC approaches address these issues by optimizing over a set of control policies rather than control actions. However, determining a suitable control policy beforehand is often prohibitively difficult [27]. Tube-based RMPC and stochastic MPC (SMPC) have been employed for legged locomotion using LIP models to accommodate uncertainties [29], [30]. Tube-based MPC involves the computation of robust invariant (RI) tubes and pre-stabilizing feedback controllers that can be computed offline for linear time-invariant systems. SMPC also leverages offline computation for the pre-stabilizing controller and chance constraints. For high-dimensional and nonlinear template models like SRB or even linear time-varying systems arising from their successive linearizations, the computation of RI tubes and pre-stabilizing linear feedback controllers should be done online, making these strategies computationally expensive. Alternative techniques [31] have used the Gaussian mixture model to learn the constraints to mitigate the gap.

In this work, we aim to answer the following *questions*: 1) How to develop a computationally tractable RMPC framework to steer template models subject to a set of unmodeled dynamics? 2) How to translate the optimal reduced-order trajectories to full-order models to bridge the gap between reduced- and full-order models? and 3) How to learn the set of unmodeled dynamics to be used in the RMPC framework?

B. Objectives and Contributions

The *objectives* and *contributions* of this paper are as follows. The paper presents a computationally tractable RMPC framework, based on convex QP, to steer template models subject to a set of modeling and abstraction uncertainties. The RMPC framework considers possible disturbances from the uncertainty set and optimizes the sum of performance criteria. We then present a hierarchical nonlinear control algorithm for real-time planning and control of quadrupedal robots. At the higher level, the RMPC framework is applied to the SRB dynamics to generate optimal ground reaction forces (GRFs) for the reduced-order model subject to a convex set of uncertainties. The proposed robust control formulation allows the integration of the RMPC framework with a fully-connected multilayer perceptron (MLP) network, trained using RL, to numerically compute the uncertainty set (see Fig. 1). At the lower level, a nonlinear controller, developed based on virtual constraints and QP, maps the generated desired GRFs to the full-order dynamical model while regulating some output functions for whole-body motion control. The use of virtual constraints-based controller is motivated by the successful hardware implementation on various bipedal [32]–[35] and quadrupedal [36], [37] robots and powered prosthetic legs [38],

[39]. The proposed robust control strategy can bridge the gap between reduced- and full-order models while learning the uncertainty at the trajectory planning level. The hierarchical control algorithm is finally validated via a set of extensive numerical simulations and experiments for the blind and robust locomotion of the A1 quadrupedal robot on different terrains and at different speeds.

The current work differs from our previous work [36] in that it only considers low-level and QP-based nonlinear controllers for quadrupedal locomotion while addressing their continuous differentiability, but not the RMPC framework and the proposed hierarchical control algorithm. The work is also different from [16] in that it does not address the robust planning subject to modeling uncertainties.

II. Robust MPC Formulation

The objective of this section is to present the high-level RMPC framework for steering template models subject to a set of uncertainties. In this paper, we consider the SRB model, but the results we present can be extended to other template-based models as well. The equations of motion for the SRB dynamics can be expressed as follows:

$$\Sigma : \begin{cases} \ddot{r} = \dfrac{f^{\text{net}}}{m} - g_0 \\ \dot{R} = R\, S(\omega) \\ I\, \dot{\omega} + S(\omega)\, I\, \omega = R^\top \tau^{\text{net}}, \end{cases} \tag{1}$$

where m is the total mass, g_0 is the constant gravitational vector, I is the body inertia, $r \in \mathbb{R}^3$ denotes the center of mass (COM) position of the robot in an inertial world frame, $R \in \text{SO}(3)$ represents the rotation matrix of the body frame with respect to the world frame, and ω denotes the angular velocity in the body frame. In our notation, $S : \mathbb{R}^3 \to \mathfrak{so}(3)$ is the skew-symmetric operator. In addition, f^{net} and τ^{net} represent the net force and torque, generated by the legs, acting on the COM, respectively, that is, $f^{\text{net}} := \sum_{j\in\mathcal{C}} f_j$ and $\tau^{\text{net}} := \sum_{j\in\mathcal{C}} S(r_j)\, f_j$, where $j \in \mathcal{C}$ denotes the foot index, $\mathcal{C}$ is the set of contacting points with the ground, $f_j \in \mathbb{R}^3$ represents the GRF at the foot $j \in \mathcal{C}$, and $r_j \in \mathbb{R}^3$ denotes the distance between the foot j and the COM. The model is valid if $f_j \in \mathcal{FC}$ for every contacting leg $j \in \mathcal{C}$, where $\mathcal{FC}$ represents the linearized friction cone.

Linearizing the SRB dynamics (1) around the current states at discrete time k via techniques such as the variational-based linearization [7] results in

$$x_{k+1} = A_k\, x_k + B_k\, u_k + E\, w_k, \quad k = 0, 1, \cdots, \tag{2}$$

where $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$, $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$, $w \in \mathcal{W} \subset \mathbb{R}^{n_w}$ denote the reduced-order states, control inputs (i.e., GRFs), and *unknown* uncertainties, respectively, for some positive integers n_x, n_u, and n_w. Here, w_k represents the uncertainties arising from abstraction and ignoring the full-order dynamics such as leg dynamics or compliant elements at the leg ends. The inclusion of w_k in (2) can help us to have a better mapping between the actual GRFs u_k and the states x_k in real world. In (2), (A_k, B_k) represents the linearized dynamics around the current state with E being the uncertainty distribution matrix. In our notation, $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$ denote the state space, admissible set of controls, and set of uncertainties, all taken as compact and convex sets containing the origin. Similar to [24], $\mathcal{W}$ is assumed to be chosen as the convex hull of some *known* vertices, that is,

$$\mathcal{W} := \text{co}\left\{ w^\ell \,|\, \ell \in \mathcal{I} \right\} \tag{3}$$

for some *finite* set of vertices indexed by $\ell \in \mathcal{I}$. In particular, w_k in (2) is an unknown signal that can be represented as a convex combination of the known vertices w^ℓ, $\ell \in \mathcal{I}$. Section IV will show how to numerically compute the uncertainty set $\mathcal{W}$ via a trained MLP.

The planning problem consists of designing an RMPC algorithm to steer (2) from an initial state to a final one in the presence of $w \in \mathcal{W}$ and subject to the feasibility conditions $x \in \mathcal{X}$ and $u \in \mathcal{U}$. To address this problem, we consider the following set of predictions at time k

$$x^\ell_{k+i+1|k} = A_k\, x^\ell_{k+i|k} + B_k\, u^\ell_{k+i|k} + E\, w^\ell\, \delta(i) \tag{4}$$

for all future times $i = 0, 1, ..., N-1$ and all realization indices $\ell \in \mathcal{I}$, where N represents the control horizon. Here, $w^\ell\, \delta(i)$ denotes a possible realization of the uncertainty with the index $\ell \in \mathcal{I}$, and $\delta(i)$ represents the discrete-time impulse (i.e., sample) function. Further, $x^\ell_{k+i|k}$ and $u^\ell_{k+i|k}$ represent the predicted state and control sequences associated with the realization $\ell \in \mathcal{I}$ and the initialization rule of $x^\ell_{k|k} = x_k$.

***Remark** 1:* In the prediction model (4), the uncertainty is considered as $w^\ell\, \delta(i)$ which is nonzero at $i = 0$ and zero for all $i > 0$. In particular, at every time sample k, the RMPC algorithm is aware of possible uncertainties that can happen at the actual time k, but it does not consider further uncertainties for predicted future times $k + i$ with $i > 0$. This assumption will reduce the number of possible branches to realize the predicted future states and control inputs in the RMPC algorithm. Hence, it will reduce the associated computational burden. This notion will be clarified more in Section V-A.

We next present the following real-time and convex QP

$$\begin{aligned} \min_{\boldsymbol{u}} \quad & \sum_{\ell\in\mathcal{I}} \left\{ p\left(x^\ell_{k+N|k}\right) + \sum_{i=0}^{N-1} \mathcal{L}\left(x^\ell_{k+i|k}, u^\ell_{k+i|k}\right) \right\} \\ \text{s.t.} \quad & x^\ell_{k+i|k} \in \mathcal{X}, \qquad i = 1, \cdots, N-1,\ \forall \ell \in \mathcal{I} \\ & u^\ell_{k+i|k} \in \mathcal{U}, \qquad i = 0, \cdots, N-1,\ \forall \ell \in \mathcal{I} \\ & x^\ell_{k+N|k} \in \mathcal{X}_f, \qquad \forall \ell \in \mathcal{I} \\ & u^{\ell_1}_{k|k} = u^{\ell_2}_{k|k}, \qquad \forall \ell_1, \ell_2 \in \mathcal{I}, \end{aligned} \tag{5}$$

where $\boldsymbol{u} := \text{col}\{\boldsymbol{u}^\ell \,|\, \ell \in \mathcal{I}\}$ represents the sequence of optimal control inputs over all realizations $\ell \in \mathcal{I}$ and $\boldsymbol{u}^\ell := \text{col}\{u^\ell_{k+i|k} \,|\, i = 0, 1, \cdots, N-1\}$ denotes the sequence of optimal controls for one particular realization ℓ. In our notation, "col" represents the column operator. Here, $p(x^\ell_{k+N|k})$ and $\mathcal{L}(x^\ell_{k+i|k}, u^\ell_{k+i|k})$ are the terminal and stage costs, respectively, defined as $p(x) := \|x - x_{\text{des}}\|^2_P$ and $\mathcal{L}(x, u) := \|x - x_{\text{des}}\|^2_Q + \|u\|^2_{\hat{R}}$ for some positive definite matrices P, Q, and $\hat{R}$, and some desired trajectory $x_{\text{des}}(k)$. In addition, $\mathcal{X}_f \subset \mathcal{X}$ is a convex and compact terminal region containing the origin. The last constraint in (5) is referred to

as the "causality constraint" to restrict the freedom on the control sequences $\boldsymbol{u}^\ell$ at the first time sample. In particular, the optimal value of the first control input to be applied to the real-world system (i.e., $u_k = u^\ell_{k|k} = \pi(x_k)$) will not depend on the realization index ℓ. We remark that the optimal control problem (5) optimizes over all possible sequences of control inputs for different realizations while satisfying the feasibility and causality conditions. Unlike [24] that uses a min-max optimization problem, we robustify against disturbances that happen only at the current time step. This will be clarified more in Section V-A. In case the uncertainty set is taken as the trivial set of $\mathcal{W} = \{0\}$, the RMPC framework is reduced to the nominal MPC.

III. Low-Level Nonlinear Controller

The objective of this section is to present the low-level nonlinear control algorithm, based on virtual constraints and QPs. The low-level controller maps the desired GRFs, generated by the high-level RMPC, to the joint-level torques while considering the nonlinear full-order model for output regulation. The full-order and floating-based dynamical model of the robot can be described as follows:

$$D(q)\,\ddot{q} + H(q,\dot{q}) = \Upsilon\,\tau + \sum_{j\in\mathcal{C}} J_j^\top(q)\,f_j, \tag{6}$$

where $q \in \mathcal{Q} \subset \mathbb{R}^{n_q}$ represents the generalized coordinates for some n_q, $D(q) \in \mathbb{R}^{n_q\times n_q}$ denotes the mass-inertia matrix, $H(q,\dot{q}) \in \mathbb{R}^{n_q}$ represents the Coriolis, centrifugal, and gravitational terms, $\tau \in \mathbb{R}^{n_\tau}$ denotes the joint-level torques for some $n_\tau < n_q$, and $\Upsilon \in \mathbb{R}^{n_q\times n_\tau}$ represents the input distribution matrix. Further, $J_j(q)$ and f_j denote the contact Jacobian matrix and GRF at the contacting leg $j \in \mathcal{C}$, respectively. For future purposes, we define $x_f := \text{col}(q,\dot{q})$ as the full-order state vector of the system. In addition, $f := \text{col}\{f_j \,|\, j \in \mathcal{C}\}$ represents the GRF vector containing the individual GRFs at the contacting leg ends.

We next consider the following holonomic output functions, referred to as *virtual constraints* [16], to be regulated

$$y(t,q) := y_a(q) - y_{\text{des}}(t), \tag{7}$$

where $y_a(q)$ represents a set of holonomic controlled variables, and $y_{\text{des}}(t)$ denotes their desired evolution on the gait in terms of a time-based phase variable. In this paper, the controlled variables are taken as the absolute position of the COM and the base frame orientation with respect to the inertial world frame together with the Cartesian coordinates of the swing leg ends for foot placement. The desired evolution for the Cartesian coordinates of the swing leg ends is parameterized as Bézier polynomials that connect the current footholds to the upcoming ones.

Differentiating the output (7) twice along the full-order dynamics of the robot results in

$$\ddot{y} = \Psi_1(x_f)\,\tau + \Psi_2(x_f)\,f + \zeta(x_f) = -K_P\,y - K_D\,\dot{y} + v, \tag{8}$$

where $\Psi_1(x_f)$, $\Psi_2(x_f)$, and $\zeta(x_f)$ are nonlinear matrices and vectors in x_f computed based on Lie derivatives and input-output (I-O) linearization [40]. Closed-form expressions for Ψ_1, Ψ_2, ζ can be extracted similar to [17, Appendix 8.2] and are not expressed here. Here, K_P and K_D are positive definite matrices, and v is an additional variable, referred to as the slack variable, to be used later. The objective is to solve for (τ, f, v) to meet (8). For this purpose, we consider an alternative set of algebraic equations that express zero accelerations for the stance leg ends. In particular, differentiating the Cartesian coordinates of the stance leg ends results in

$$\ddot{r}_{\text{st}} = \Omega_1(x_f)\,\tau + \Omega_2(x_f)\,f + \eta(x_f) = 0, \tag{9}$$

where r_{st} represents a vector containing the Cartesian coordinates of all stance legs, and $\Omega_1(x_f)$, $\Omega_2(x_f)$, and $\eta(x_f)$ are proper matrices and vectors.

We now aim to solve for (τ, f, v) to satisfy (8) and (9) while (τ, f) being feasible, that is, $f_j \in \mathcal{FC}$ for all $j \in \mathcal{C}$ and $\tau \in \mathcal{T}$, where $\mathcal{T}$ represents a convex and compact set of admissible torques. For this purpose, we set up the following real-time and strictly convex QP to be solved at 1kHz

$$\begin{aligned}
\min_{(\tau,f,v)}\ & \frac{\gamma_1}{2}\|\tau\|^2 + \frac{\gamma_2}{2}\|f - f_{\text{des}}\|^2 + \frac{\gamma_3}{2}\|v\|^2 \\
\text{s.t.}\ & \Psi_1(x_f)\,\tau + \Psi_2(x_f)\,f + \zeta(x_f) = -K_P\,y - K_D\,\dot{y} + v \\
& \Omega_1(x_f)\,\tau + \Omega_2(x_f)\,f + \eta(x_f) = 0 \\
& \tau \in \mathcal{T}, \quad f_j \in \mathcal{FC}, \quad \forall j \in \mathcal{C}.
\end{aligned} \tag{10}$$

Here, γ_1, γ_2, and γ_3 are positive weighting factors. The slack variable v is added to (8) to ensure the feasibility of the QP. To minimize the effect of the slack variable on the output dynamics (8), a penalty term $\frac{\gamma_3}{2}\|v\|^2$ is added to the cost function via a large weighting factor γ_3. The other two terms in the cost function try to minimize the 2-norm of the force tracking error $f - f_{\text{des}}$ while having minimum-power torques. This can help to bridge the gap between the planned (i.e., desired) GRFs (f_{des}), computed based on the template model and RMPC, and the actual ones (f). The optimal torques from this low-level nonlinear controller are applied to the full-order dynamics and are denoted by $\tau = \Gamma(t, x_f)$ (see Fig. 1).

IV. Learning to Close the Gap

The objective of this section is to numerically compute the vertices of the uncertainty set $\mathcal{W}$ in (3) to be used in the RMPC framework (see Fig. 1 again). The problem of computing the uncertainty set $\mathcal{W}$ is posed as a sequential decision-making process that computes the mapping from the reduced-order states x_k to the vertices of the set $\mathcal{W} \subset \mathbb{R}^{n_w}$. This can be mathematically represented using a Markov Decision Process (MDP). We then use a model-free RL technique [42] to train a fully-connected MLP network that computes the vertices of the uncertainty set.

We implement our hierarchical control algorithm in-the-loop with the full-order simulation model of the quadrupedal robot to train the MLP in an offline manner. Here, we consider the problem of locomotion over rough terrains and represent the sequential decision-making process in discrete time. At every time t, the MDP obtains an observation and performs an action while achieving a scalar reward $\mathcal{R}_t$. The actions of the MDP represent the vertices of uncertainty set $\mathcal{W}$, that is,

$$w^l_k = \Phi^\ell(x_k), \quad \ell \in \mathcal{I}, \tag{11}$$

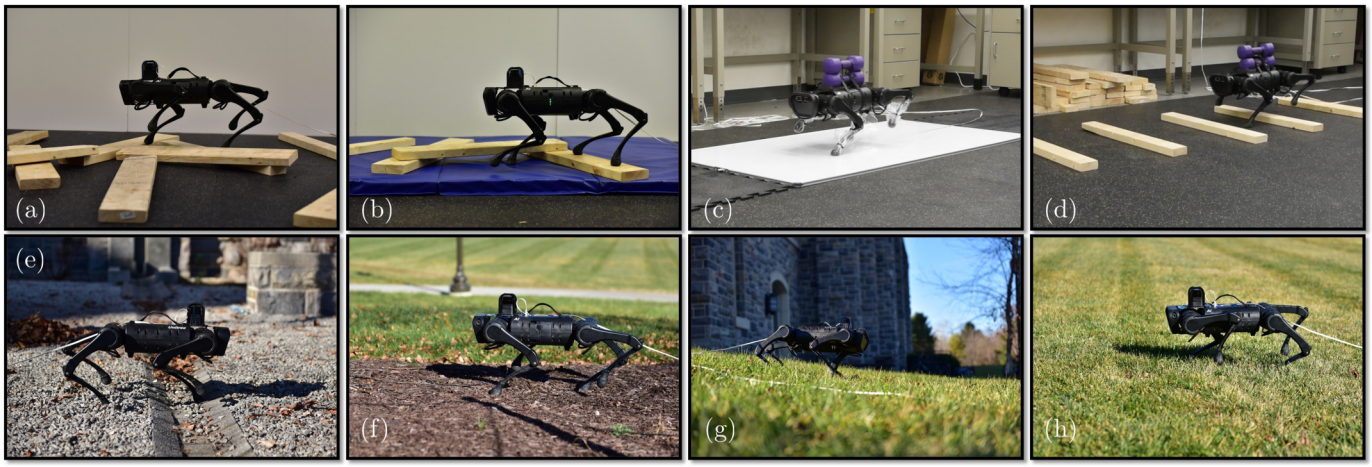

Fig. 2. Indoor and outdoor experiments to validate the proposed hierarchical control algorithm on different terrains. (a) Blind locomotion of the A1 robot on uneven terrain with wooden blocks at the speed of 0.5 (m/s). (b) Locomotion on a compliant surface with unknown blocks. (c) Locomotion on a slippery surface with a payload of 4.54 (kg) (36% of the robot's weight). (d) Locomotion on blocks with a payload of 4.54 (kg). (e) Locomotion on uneven gravel terrain. (f) Locomotion on mulch. (g) Locomotion on a slope. (h) Locomotion on the grass at the speed of 1 (m/s). Videos are available online [41].

where $\Phi^\ell(\cdot)$ denotes the trained NN, and x_k (reduced-order states) represents the observables for the MDP. Here, we consider one augmented MLP that computes all of the vertices, that is, $\Phi(\cdot) := \text{col}\{\Phi^\ell(\cdot) \,|\, \ell \in \mathcal{I}\}$. For the SRB dynamics expressed in (1), the observables for the MDP are taken as the z-component of the COM position (r_z), the base-frame orientation (R), the linear and angular velocities ($\dot{r}$ and ω), and the relative positions of feet from the COM (r_j). We aim to find an optimal policy $\Phi^\star$ that maximizes the following expected sum of rewards over T samples

$$\Phi^\star = \arg\max_{\Phi} \mathbb{E}\left\{\sum_{t=0}^{T-1} \lambda^t \mathcal{R}_t\right\}, \tag{12}$$

where $\lambda \in (0, 1)$ represents the discount factor.

In this paper, we employ the Proximal Policy Optimization (PPO) algorithm [43] to solve (12) and use the A1 quadrupedal robot, developed by Unitree, as our test-bed. Similar to [44], the reward function at time t is chosen as

$$\mathcal{R}_t = \alpha_v \, \|\dot{r}_x\|^2 - \alpha_\tau \, \|\tau\|^2, \tag{13}$$

where $\dot{r}_x$ and τ denote the forward velocity of the COM (i.e., along the x-axis) and torques, respectively, and α_v and α_τ represent positive weighting factors. This work approximates both the policy and value functions using MLPs, each with two hidden layers of 128 neurons. We also choose the discounting factor as $\lambda = 0.998$ with $\alpha_v = 1$ and $\alpha_\tau = 4 \times 10^{-5}$. Details of the numerical simulations together with the controller and training parameters will be presented in Section V.

V. Numerical and Experimental Results

This section aims to numerically and experimentally validate the proposed control algorithm for the robust and blind locomotion of the A1 quadrupedal robot on different terrains and at different speeds.

A. Controller Synthesis

The A1 robot is a torque-controlled robot with 18 Degrees of Freedom (DOFs) and 12 actuators, weighing 12.45 (kg) and standing up to 0.26 (m) (see Fig. 2). Six DOFs describe the unactuated and absolute position and orientation (i.e., Euler angles) of the robot's body with respect to the inertial world frame. The remaining 12 DOFs describe the actuated joints in legs. In particular, each leg has a 2-DOF hip joint (hip roll and hip pitch) and an additional 1-DOF knee joint (knee pitch). We use RaiSim [45] and implement the RL framework outlined in [44] for training of the MLP. The proposed high-level RMPC and the low-level QP are solved online using qpSWIFT [46] at 160Hz and 1kHz, respectively. In the RMPC framework, we consider the linearized SRB dynamics (1) with 12 states for trotting and subject to a set of unmodeled dynamics (i.e., $E\,w$) that appear as additional forces/torques in the velocity dynamics (i.e., $\ddot{r}$ and $\dot{\omega}$) (see (2)). More specifically, the uncertainty vector w is taken as 6-dimensional arising from abstraction (e.g., ignoring the leg dynamics) to have a better mapping from the actual GRFs to the SRB dynamics. We then take $E = [\mathbb{I}_{6\times6}; \mathbb{I}_{6\times6}] \in \mathbb{R}^{12\times6}$ as a selection matrix composed of zeros and ones to distribute the vertices w across the realization of the RMPC. We remark that one can alternatively consider the dynamics as $x_{k+1} = A_k\, x_k + B_k\, u_k + w_k$. However, our choice of the selection matrix E reduces the dimension of the vertices of $\mathcal{W}$ from 12 to 6, and hence, it reduces the hyperparameters for training the MLP in (11).

To reduce the computational burden associated with the RMPC algorithm (5), we consider the uncertainty set $\mathcal{W} \subset \mathbb{R}^6$ as a convex hull of two vertices, that is, $\mathcal{W} = \text{co}\{w^\ell \,|\, \ell \in \mathcal{I}\}$ and $\mathcal{I} = \{1, 2\}$ (See Section V-C for a comparative study on the effects of the number of vertices for $\mathcal{W}$). The control horizon for the RMPC problem is chosen as $N = 7$ discrete sample times. The resultant RMPC and QP-based low-level controller will be solved on an off-board laptop with i7-1185G7 running at 3.00 GHz in experiments of Section V-C. Under nominal conditions, the computation time for the RMPC and low-level controller is approximately 4.2 (ms) and 0.22 (ms), respectively. As discussed in Remark 1, we only consider the branching in the state trajectories at the first time sample. More specifically, 2 different sets of state trajectories, referred to as realizations, are considered in the RMPC framework (5). The min-max technique of [24], how-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3176105, IEEE Robotics and Automation Letters

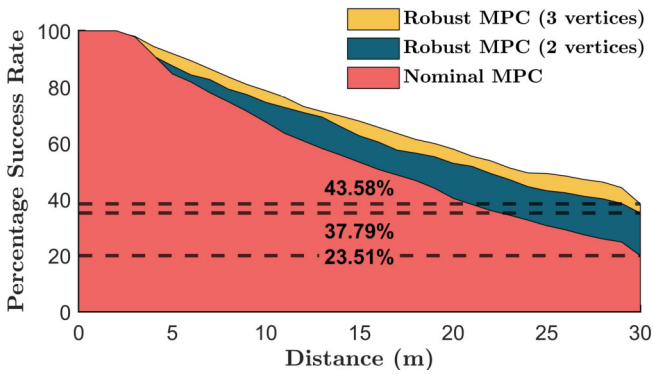

Fig. 3. The percentage of the success rate of RMPC with three and two vertices and the nominal MPC over 550 randomly generated uneven terrains. The dashed line represents the success rate across the length of the terrain.

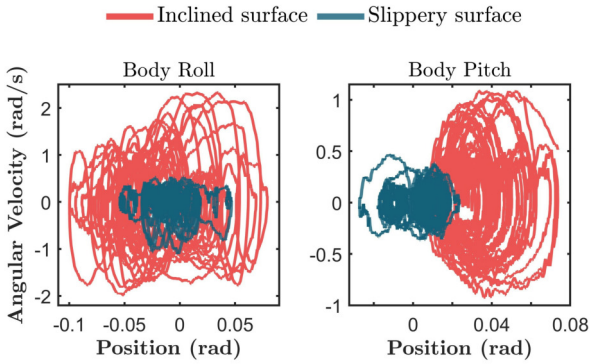

Fig. 4. Phase portrait of the unactuated DOFs (roll and pitch). The figure depicts robust stability in the phase portraits for two different experiments, including outdoor locomotion on inclined terrain (Fig. 2(g)) and indoor locomotion on a slippery surface with a payload of 4.54 (kg) (Fig. 2(c)).

ever, considers all possible branches that can happen at every time sample within the control horizon and then minimizes the worst-case scenario. This approach would then result in $2^7 = 128$ different realizations, which is not computationally tractable for real-time planning. Unlike [24], the proposed RMPC approach does not guarantee the recursive feasibility. In the rare events in which the QP becomes infeasible, we terminate the QP solver at a maximum iteration number and use the solution. The feasibility set $\mathcal{X}$ and the terminal set $\mathcal{X}_f$ are chosen as hypercubes in $\mathbb{R}^{12}$ around the desired trajectory. The feasibility set $\mathcal{U}$ is then chosen as the linearized friction cone with the friction coefficient of $\mu = 0.6$.

For the low-level controller, the weighting factors are taken as $\gamma_1 = 1$, $\gamma_2 = 10^3$, and $\gamma_3 = 10^6$ with the same friction coefficient used for the high-level RMPC. In this work, we make use of a time-based switching approach for impact detection. In addition, we employ Raibert's heuristic [47, Eq. (2.4), pp. 46] for footstep planning.

B. Training of the MLP

The training of the MLP network is performed offline using numerical simulations with the learning environment provided by RaiSim [45]. The objective of training the MLP is to learn the gap between reduced- and full-order models and also to aid in the locomotion of the quadrupedal robot on rough terrains by computing the appropriate uncertainty set $\mathcal{W}$. The simulation and training environments consist of a single fixed rough terrain composed of rigid blocks placed at regular intervals. The use of a single training environment is motivated by the following justification. The QP-based virtual constraints controller employed at the low level of the hierarchical control algorithm can result in stable locomotion patterns on flat terrains, as studied in our previous work [36]. The integration of the low-level controller with the high-level RMPC algorithm improves the robust stability of gaits over different sets of terrains. Unlike other *model-free* techniques that use NNs to learn the entire whole-body trajectory planning from scratch, our proposed MLP layer only computes the appropriate set of uncertainties to be used in the RMPC framework. Hence, the combination of the MLP layer with the *model-based* RMPC and the low-level nonlinear controller does not require training on a large set of randomly generated terrains as indicated by the experimental results in Section V-C.

Each block in the fixed terrain is 5 (cm) (20% of the robot's nominal height) tall and spaced at regular intervals of 5 (cm). The robot is commanded by the MPC and low-level controller to perform a symmetrical forward trot gait at 0.5 (m/s). Each episode runs for four seconds or until it meets the termination criterion. Termination is encountered when the rigid body links connecting the knee with the toe hit the ground. The learning rate is set to 5×10^{-4}, and the entire training roughly takes 12 hours on a Desktop PC equipped with an Intel Xeon W-2125 processor. We use two hidden layers with 128 neurons in each of the MLPs of the policy and value function approximations with a Rectified Linear Unit (ReLU) as the activation function.

C. Numerical and Experimental Validation and Discussion

The objective of this section is to numerically and experimentally validate the proposed control algorithm for the robust and blind locomotion of the A1 robot on different sets of terrains and at different speeds.

To study the effects of the number of vertices on the uncertainty set $\mathcal{W}$, we conduct two numerical experiments, each with a different set of vertices. In particular, we choose two and three vertices for the uncertainty set and train the corresponding MLP. Each of the RMPC algorithms takes 12 and 18 hours to learn the uncertainty set. We then quantify the performance of the two developed RMPC algorithms with the nominal MPC. More specifically, we randomly generate 550 heightmaps, each with a different distribution of blocks. The blocks, each 5 (cm) in height, are distributed asymmetrically and randomly over a terrain of 30 (m) length ($\approx$ 85 robot's body length). Figure 3 illustrates the percentage of the success rate of each of the RMPC algorithms and the nominal MPC on 550 randomly generated terrains over the traveled distance. Here, failure is determined based on the termination criterion outlined in Section V-B. As depicted in Fig. 3, the success rate of RMPC with three and two vertices and the nominal MPC across the entire length of the terrain is 43.58%, 37.79%, and 23.51%, respectively. We remark that for the experiments on the real robot, we choose the RMPC with two vertices as the difference in success rates between the two RMPC algorithms is not significant. Furthermore, the computational time taken by RMPC with three vertices (10.9 (ms)) is significantly higher than the RMPC with two vertices (4.2 (ms)) and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3176105, IEEE Robotics and Automation Letters

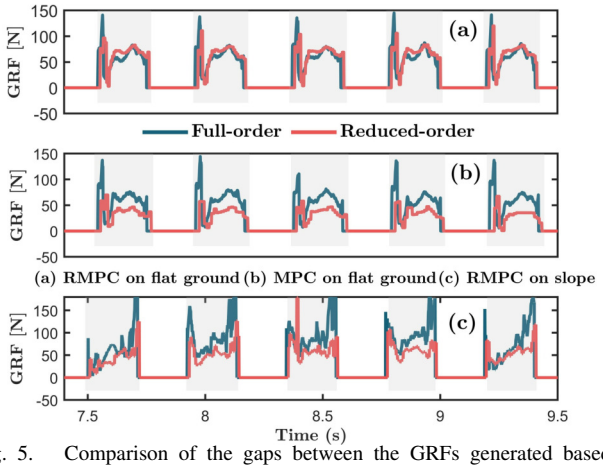

Fig. 5. Comparison of the gaps between the GRFs generated based on the reduced- and full-order models. Plot of desired and actual GRFs (the z component) with the high-level RMPC algorithm and the low-level QP-based nonlinear controller on flat terrain (a) and on a slope (c). (b) Plot of the desired and actual GRFs with the hierarchical control algorithm in which the high-level RMPC is replaced by a nominal MPC for experiments on flat terrain.

nominal MPC (1.24 (ms)), justifying our choice of $\mathcal{W}$ with two vertices. The animations of the representative simulations are available online [41].

As shown in Fig. 2, we consider four different subsets of indoor investigations, including locomotion a) on uneven terrain with a random configuration of wooden blocks, b) on a compliant surface covered with wooden blocks, c) with a payload of 4.54 (kg) (36% of the robot's weight) on a slippery surface, covered by a cooking spray, and d) with a payload of 4.54 (kg) on wooden blocks. The outdoor experiments then include locomotion on e) uneven gravel terrain, f) mulch, g) a slope, and h) the grass. For some of these experiments, we consider locomotion at different speeds of 0.1 (m/s), 0.5 (m/s), 0.75 (m/s), and 1 (m/s). We observe that the proposed hierarchical control algorithm can result in robustly stable gaits on these different sets of terrains. Figure 4 shows the phase portraits of the robot's roll and pitch motions during two experiments. The first experiment involves locomotion in an outdoor environment on an inclined lawn with an approximate inclination of $(10^\circ - 15^\circ)$ (Fig. 2(g)). The second experiment involves locomotion in an indoor environment on a slippery surface (Fig. 2(c)). The slippery environment is created with the use of a cooking spray. For this experiment, the coefficient of friction is reduced to 0.3 in both the MPC and the low-level controller. Videos of all experiments can be found online [41].

Figure 5 studies the gap between the reduced- and full-order models of locomotion with and without the proposed hierarchical control algorithm. Here, we plot the time evolution of the desired and actual GRFs at the front left leg for experiments on flat ground and trotting backward on a downhill slope. The desired GRF is the optimal control input computed for the SRB dynamics via the MPC algorithm. The actual GRF is then computed according to the full-order dynamics via the QP-based nonlinear controller in (10). Figures 5(a) and 5(c) compare the vertical components of the desired and actual GRFs with the proposed RMPC framework. In Fig. 5(b), the RMPC framework at the higher level of the control algorithm is replaced with a nominal MPC. It is clear that there is a

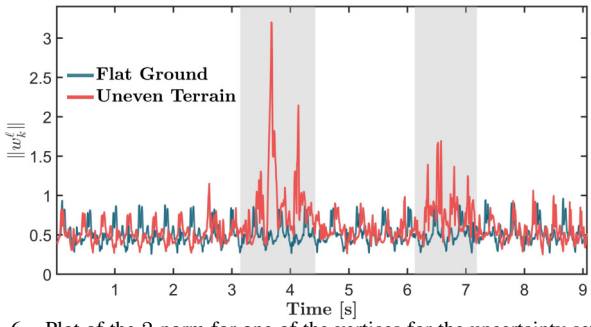

Fig. 6. Plot of the 2-norm for one of the vertices for the uncertainty set (i.e., $\|w_k^\ell\|$) versus time. The plot depicts the contribution of the trained MLP for locomotion on a flat ground and locomotion on an uneven terrain with two blocks. The gray portions indicate the stepping on blocks.

significant reduction between the GRFs computed based on the reduced- and full-order models compared to Fig. 5(b). In addition, it can be seen from Fig. 5(c), that bridging the gap occurs for experiments involving locomotion on a slope.

To study the contribution of the trained MLP in the RMPC framework, we plot the 2-norm of the upper-bound vertex of the uncertainty set versus time in Fig. 6. Here, we consider two indoor experiments, including the nominal locomotion on flat ground and the locomotion on two wooden blocks. From Fig. 6, it can be observed that $\|w_k^\ell\|$ spikes during stepping on blocks and decreases as the robot continues to step forward. During the steady-state, $\|w_k^\ell\|$ reaches periodic behavior that is smaller than the spikes for uneven terrains.

VI. Conclusion and Future Work

This paper presented a hierarchical control algorithm for the real-time motion planning and control of quadrupedal robots while bridging the gap between reduced- and full-order models. At the higher level, a computationally tractable RMPC framework is developed based on QPs to steer reduced-order locomotion models subject to a convex set of uncertainties arising from abstraction and unmodeled dynamics. In particular, the RMPC algorithm is applied to the SRB dynamics to generate optimal GRFs. At the lower level, a nonlinear controller, based on I-O linearization and QPs, is developed to map the optimal reduced-order GRFs to the full-order model while imposing virtual constraints for whole-body motion control. The proposed RMPC framework allows the integration of the hierarchical controller with RL techniques to train an MLP to compute the vertices of the uncertainty set numerically. The proposed hierarchical control algorithm is finally validated numerically and experimentally for robust and blind locomotion of the A1 quadrupedal robot on different indoor and outdoor terrains and at different speeds. The numerical analysis of the RMPC suggests significant improvement in the performance of the rough terrain locomotion compared to the nominal MPC. Our experimental studies indicate a significant reduction in the gap between the reduced- and full-order models by comparing the desired and actual GRFs computed by the proposed high- and low-level controllers.

The current work considered uncertainty in the template model dynamics. For future research, we will investigate the incorporation of uncertainties in kinematic constraints. We will also explore alternative reward functions to mitigate the gap between reduced- and full-order models.

REFERENCES

[1] R. Full and D. Koditschek, "Templates and anchors: Neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[2] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 1405–1406.

[3] 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.

[4] Y. Ding, A. Pandala, and H.-W. Park, "Real-time model predictive control for versatile dynamic motions in quadrupedal robots," in *2019 IEEE International Conference on Robotics and Automation*, 2019, pp. 8484–8490.

[5] 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*, Oct 2018, pp. 2245–2252.

[6] M. Chignoli and P. M. Wensing, "Variational-based optimal control of underactuated balancing for dynamic quadrupeds," *IEEE Access*, vol. 8, pp. 49 785–49 797, 2020.

[7] 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, no. 4, pp. 1154–1171, 2021.

[8] R. J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, and J. Pratt, "Walking stabilization using step timing and location adjustment on the humanoid robot, Atlas," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 667–673.

[9] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on Capture Point dynamics," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 4420–4427.

[10] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A step toward humanoid push recovery," in *2006 IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.

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

[12] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2018, pp. 1–9.

[13] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *2016 IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 558–564.

[14] R. Grandia, F. Farshidian, A. Dosovitskiy, R. Ranftl, and M. Hutter, "Frequency-aware model predictive control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1517–1524, 2019.

[15] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *2017 IEEE International Conference on Robotics and Automation*, 2017, pp. 93–100.

[16] K. Akbari Hamed, J. Kim, and A. Pandala, "Quadrupedal locomotion via event-based predictive control and QP-based virtual constraints," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4463–4470, 2020.

[17] J. Kim and K. Akbari Hamed, "Cooperative locomotion via supervisory predictive control and distributed nonlinear controllers," *ASME Journal of Dynamic Systems, Measurement, and Control*, 11 2021.

[18] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 8352–8358.

[19] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 1623–1630.

[20] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *2018 IEEE International Conference on Robotics and Automation*, 2018, pp. 4399–4406.

[21] S. Kolathaya, J. Reher, A. Hereid, and A. D. Ames, "Input to state stabilizing control Lyapunov functions for robust bipedal robotic locomotion," in *American Control Conference*, 2018, pp. 2224–2230.

[22] J.-H. Hours, M. N. Zeilinger, R. Gondhalekar, and C. N. Jones, "Constrained spectrum control," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1969–1974, 2014.

[23] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[24] P. Scokaert and D. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 8, pp. 1136–1142, 1998.

[25] P. J. Campo and M. Morari, "Robust model predictive control," in *1987 American control conference*, 1987, pp. 1021–1026.

[26] Z. Q. Zheng and M. Morari, "Robust stability of constrained model predictive control," in *1993 American Control Conference*, 1993, pp. 379–383.

[27] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[28] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[29] N. A. Villa and P.-B. Wieber, "Model predictive control of biped walking with bounded uncertainties," in *IEEE-RAS International Conference on Humanoid Robotics*, 2017, pp. 836–841.

[30] A. Gazar, M. Khadiv, A. Del Prete, and L. Righetti, "Stochastic and robust MPC for bipedal locomotion: A comparative study on robustness and performance," in *IEEE-RAS International Conference on Humanoid Robots*, 2021, pp. 61–68.

[31] J. Carpentier and N. Mansard, "Multicontact locomotion of legged robots," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, 2018.

[32] A. E. Martin, D. C. Post, and J. P. Schmiedeler, "The effects of foot geometric properties on the gait of planar bipeds walking under HZD-based control," *The International Journal of Robotics Research*, vol. 33, no. 12, pp. 1530–1543, 2014.

[33] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for achieving stable, efficient and fast bipedal walking on MABEL," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, Aug. 2011.

[34] X. Da and J. Grizzle, "Combining trajectory optimization, supervised machine learning, and model structure for mitigating the curse of dimensionality in the control of bipedal robots," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1063–1097, 2019.

[35] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 370–387, 2018.

[36] R. T. Fawcett, A. Pandala, A. D. Ames, and K. Akbari Hamed, "Robust stabilization of periodic gaits for quadrupedal locomotion via QP-based virtual constraint controllers," *IEEE Control Systems Letters*, pp. 1736–1741, 2021.

[37] W.-L. Ma, N. Csomay-Shanklin, S. Kolathaya, K. Akbari Hamed, and A. D. Ames, "Coupled control Lyapunov functions for interconnected systems, with application to quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3761–3768, 2021.

[38] R. Gregg and J. Sensinger, "Towards biomimetic virtual constraint control of a powered prosthetic leg," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 1, pp. 246–254, Jan 2014.

[39] H. Zhao, J. Horn, J. Reher, V. Paredes, and A. D. Ames, "Multicontact locomotion on transfemoral prostheses via hybrid system models and optimization-based control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 502–513, 2016.

[40] A. Isidori, *Nonlinear Control Systems*. Springer; 3rd edition, 1995.

[41] Robust mpc for quadrupedal locomotion. [Online]. Available: https://youtu.be/uK4V1bKJSzk.

[42] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[44] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, 2020.

[45] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, April 2018.

[46] A. G. Pandala, Y. Ding, and H. Park, "qpSWIFT: A real-time sparse quadratic program solver for robotic applications," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3355–3362, Oct 2019.

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