Toward a Data-Driven Template Model for Quadrupedal Locomotion

Randall T. Fawcett[1], Kereshmeh Afsari[2], Aaron D. Ames[3], and Kaveh Akbari Hamed[1]

Abstract—**This work investigates a data-driven template model for trajectory planning of dynamic quadrupedal robots. Many state-of-the-art approaches involve using a reduced-order model, primarily due to computational tractability. The spirit of the trajectory planning approach in this work draws on recent advancements in the area of behavioral systems theory. Here, we aim to capitalize on the knowledge of well-known template models to construct a data-driven model, enabling us to obtain an information rich reduced-order model. In particular, this work considers input-output states similar to that of the single rigid body model and proceeds to develop a data-driven representation of the system, which is then used in a predictive control framework to plan a trajectory for quadrupeds. The optimal trajectory is passed to a low-level and nonlinear model-based controller to be tracked. Preliminary experimental results are provided to establish the efficacy of this hierarchical control approach for trotting and walking gaits of a high-dimensional quadrupedal robot on unknown terrains and in the presence of disturbances.**

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

I. Introduction

MANY of the current state-of-the-art approaches for planning or controlling legged robots rely on a reduced-order (i.e., template) model of the robot [1]. This is done to gain real-time computational tractability while retaining the dominant traits of the nonlinear dynamics by providing a low-dimensional approximation of the full-order dynamics. This work aims to construct a template model based on data obtained during locomotion to provide a mapping from some desired inputs to some desired outputs. This is intended to allow one to construct a reduced-order model without explicitly having access to model parameters while also potentially encapsulating some of the rich nonlinear dynamics. This additionally removes a layer of abstraction introduced by linearizing a physics-based template model.

A. Reduced-Order Models and Motivation

Many works consider the combination of reduced-order models and model predictive control (MPC) frameworks for trajectory planning of legged robots. Among the most well-studied template models is the Linear Inverted Pendulum (LIP) model [2]. In its most basic form, the LIP model is restrictive since it considers the body as a point mass and requires the center of pressure (COP) to remain within the convex hull formed by the contacting legs. The LIP model has been studied extensively, particularly on bipeds with non-trivial feet, and has been validated on several platforms for quasi-static locomotion of both bipeds and quadrupeds [2]–[6]. Largely due to the quasi-static nature of LIP-based models, many control approaches have shifted toward template models that are more directly amenable to dynamic motions. For example, the Spring-Loaded Inverted Pendulum (SLIP) model has been used successfully to produce more dynamic motions [7], [8]. However, the SLIP model still treats the body as a point mass and suffers from nonlinearity.

Another very popular method is to consider the torso as a single rigid body (SRB) being propagated through space via forces applied to the body [9]. The SRB model, combined with standard linearization and MPC techniques, has proven to be a powerful technique for stable locomotion and has been experimentally validated on several quadrupedal platforms [9]–[12]. One of the primary flaws of the SRB model is the assumption that the legs have negligible mass. While this is a reasonable assumption for small quadrupeds, it does not readily extend to robots with more massive legs. Reference [11] has used quasi-static compensation for the mass of the legs during balance control but does not consider the full-order dynamics. In addition to neglecting the legs, the successive linearization of the SRB dynamics introduces another undesirable layer of abstraction. Centroidal dynamics have also been employed, which is similar in nature to the SRB model but considers the angular momentum of the torso. Centroidal dynamics have been more intensely studied on bipeds [13], [14], but have proven to be effective for quadrupeds as well [15]. However, the centroidal model shares many of the same issues as the SRB model. Namely, the leg dynamics and inertial effects are generally neglected [16].

B. Data-Driven Approaches

Data-driven techniques are becoming increasingly important as systems become more complex and applications demand more rigorous controllers and have progressed substantially in

Manuscript received: February, 24, 2022; Revised May, 12, 2022; Accepted June, 8, 2022. This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments. The work of R. T. Fawcett and K. Afsari is supported by the National Science Foundation (NSF) under Grant 2128948. The work of K. Akbari Hamed is supported by the NSF under Grants 1923216 and 2128948. The work of A. D. Ames is supported by the NSF under Grant 1923239.

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

[2] K. Afsari is with the Myers-Lawson School of Construction, Virginia Tech, Blacksburg, VA, 24061, USA, keresh@vt.edu

[3] A. D. Ames is with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA, 91125, USA, ames@caltech.edu

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

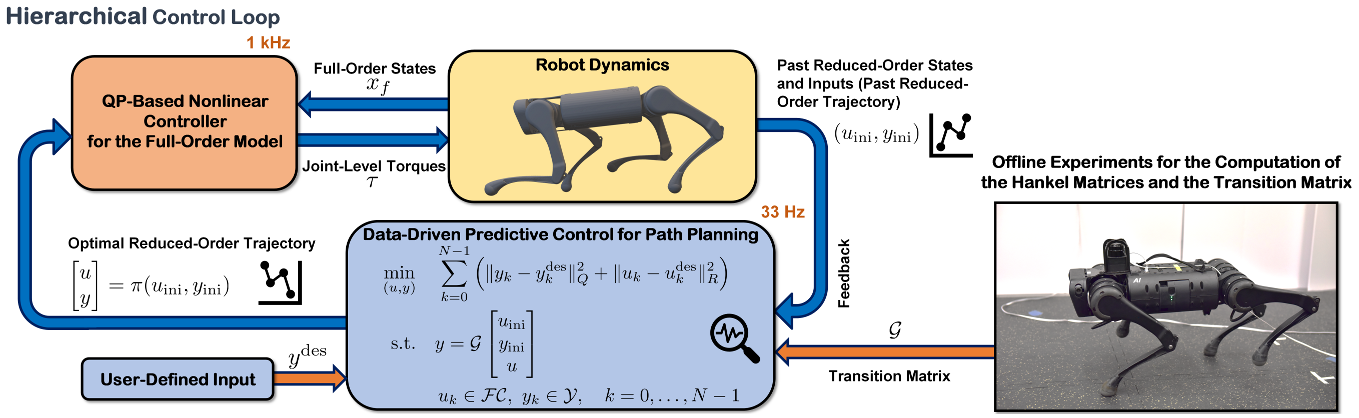
Hierarchical Control Loop
1 kHz
QP-Based Nonlinear Controller for the Full-Order Model
Full-Order States
x_f
Joint-Level Torques
τ
Robot Dynamics
Past Reduced-Order States and Inputs (Past Reduced-Order Trajectory)
$(u_{\mathrm{ini}}, y_{\mathrm{ini}})$
Offline Experiments for the Computation of the Hankel Matrices and the Transition Matrix
33 Hz
Feedback
Optimal Reduced-Order Trajectory
$\begin{bmatrix} u \\ y \end{bmatrix} = \pi(u_{\mathrm{ini}}, y_{\mathrm{ini}})$
Data-Driven Predictive Control for Path Planning
$\min_{(u,y)} \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\mathrm{des}}\|_Q^2 + \|u_k - u_k^{\mathrm{des}}\|_R^2 \right)$
$\text{s.t.} \quad y = \mathcal{G} \begin{bmatrix} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} \\ u \end{bmatrix}$
$u_k \in \mathcal{FC}, \; y_k \in \mathcal{Y}, \quad k = 0, \dots, N-1$
$\mathcal{G}$
Transition Matrix
User-Defined Input
y^{des}

Fig. 1. Overview of the proposed hierarchical control algorithm. At the high level, the data-driven predictive control generates optimal trajectories for trajectory planning of the quadrupedal robot. The optimal trajectories are then passed to a low-level and QP-based nonlinear controller for the whole-body motion control. The data-driven transition matrix is computed based on a set of offline experiments.

the last several decades [17]. Furthermore, increased dynamical complexity can require considerable expertise to obtain an accurate physics-based model. While the literature on model-free control methodologies spans many areas, including reinforcement learning [18], [19], here we focus on the use of data-driven approaches for predictive control, generally referred to as data-driven predictive control, or data-enabled predictive control (DeePC) [20]–[25]. These works stem from behavioral systems theory, used to parameterize a linear time-invariant (LTI) system in terms of its observed trajectories as opposed to physics-based dynamics [26]–[28]. Although the original theory does not directly apply to complex nonlinear systems, recent works have provided theoretical extensions to certain classes of nonlinear systems [29], implementations for stochastic and nonlinear systems [21]–[23], and linear parameter varying systems [30]. However, to the best of the authors' knowledge, there has not been an implementation for unstable hybrid dynamical models of legged robots with underactuation and unilateral constraints, which is the focus of this work. While rigorous theory has yet to be developed extending to hybrid nonlinear systems, we have observed good performance nonetheless.

C. Goals, Objectives, and Contributions

The *overarching goal* of this paper is to develop a layered control approach based on data-driven template models for real-time planning and control of dynamic quadrupedal robots. More specifically, this paper's *objectives* and *key contributions* are as follows. 1) At the higher level of the control approach, we provide a reduced-order model based on data by leveraging information about state-of-the-art template models, specifically the SRB model, which also potentially encapsulates important nonlinear information while forgoing the need for successive linearization (see Fig. 1). 2) A computationally tractable predictive controller is presented, based on a data-driven template model, for the real-time trajectory planning of high degree of freedom (DOF) quadrupeds. 3) The optimal trajectories are then passed to a low-level nonlinear controller based on virtual constraints [31] for whole-body motion control. 4) Preliminary experimental validation of the proposed layered control approach is provided on the 18-DOF quadrupedal robot A1 for a walk and trot gait and differing gait parameters for trotting. The experimental results also show robust locomotion of the A1 robot on unknown terrains and in the presence of disturbances.

II. Preliminaries

This section provides an overview of some of the pertinent components of behavioral systems theory. Behavioral systems theory provides a formal manner in which an unknown LTI system can be parameterized purely by measured trajectories of the system.

Consider the model of an LTI system with the state vector $x_k \in \mathbb{R}^n$, the input vector $u_k \in \mathbb{R}^m$, and the output vector $y_k \in \mathbb{R}^p$ for $k \in \mathbb{Z}_{\geq 0} := \{0, 1, \cdots\}$. The standard discrete-time state-space representation is described by

$$\begin{aligned} x_{k+1} &= A\, x_k + B\, u_k \\ y_k &= C\, x_k + D\, u_k, \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$ are the state space matrices which are unknown. Here we denote n, m, and p as the number of states, inputs, and outputs, respectively. Consider some $L, T \in \mathbb{N} := \{1, 2, \cdots\}$, where T is the total length of the data collected and $T \geq L$, along with some input trajectory $u^{\mathrm{d}} \in \mathbb{R}^{mT}$ composed of a sequence of collected data u_k^{d}, i.e., $u^{\mathrm{d}} := \mathrm{col}(u_0^{\mathrm{d}}, \dots, u_{T-1}^{\mathrm{d}})$. In our notation, "col" represents the column operator. As will be discussed shortly, L represents the sum of the prediction and estimation horizons. Using this trajectory, one can construct the corresponding Hankel matrix [21] as follows:

$$\mathscr{H}_L(u^{\mathrm{d}}) := \begin{bmatrix} u_0^{\mathrm{d}} & u_1^{\mathrm{d}} & \cdots & u_{T-L}^{\mathrm{d}} \\ u_1^{\mathrm{d}} & u_2^{\mathrm{d}} & \cdots & u_{T-L+1}^{\mathrm{d}} \\ \vdots & \vdots & \ddots & \vdots \\ u_{L-1}^{\mathrm{d}} & u_L^{\mathrm{d}} & \cdots & u_{T-1}^{\mathrm{d}} \end{bmatrix} \in \mathbb{R}^{mL \times (T-L+1)}. \tag{2}$$

Definition 1 ([20]): The signal u^{d} is said to be *persistently exciting* of order L if $\mathscr{H}_L(u^{\mathrm{d}})$ is full row rank, ensuring the signal contains sufficiently rich information.

Definition 2 ([20]): The sequence $\{(u_k^{\mathrm{d}}, y_k^{\mathrm{d}})\}_{k=0}^{T-1}$ is said to be a *trajectory* of the LTI system (1) if there exists an initial condition x_0 and a state sequence $\{x_k\}_{k=0}^{T}$ that meets the state and output equations in (1).

Using Definitions 1 and 2, we are now in a position to present a foundational theorem that is used to define an LTI system in terms of its trajectories.

Theorem 1: [26, Theorem 1] *Let a trajectory of an LTI system, referred to as data, be denoted by $\{(u_k^d, y_k^d)\}_{k=0}^{T-1}$. If u^d is persistently exciting of order $L+n$, then $\{(\bar{u}_k, \bar{y}_k)\}_{k=0}^{L-1}$ is a* trajectory of the system if and only if there exists $g \in \mathbb{R}^{T-L+1}$ such that

$$\begin{bmatrix} \mathscr{H}_L(u^{\mathrm{d}}) \\ \mathscr{H}_L(y^{\mathrm{d}}) \end{bmatrix} g = \begin{bmatrix} \bar{u} \\ \bar{y} \end{bmatrix}. \tag{3}$$

Theorem 1 presents a data-driven approach for characterizing trajectories of an unknown LTI system without requiring explicit system identification. This theorem will be used to synthesize a data-driven predictive control approach for real-time motion planning of legged robots in Section III.

In order to formulate the trajectory planning problem as a closed-loop data-driven predictive control approach, we will consider two different horizons as the *estimation horizon* T_{ini} and the *prediction (i.e., control) horizon* N. In particular, we assume that $L = T_{\text{ini}} + N$. Here, the estimation horizon T_{ini} can be viewed as the number of input-output (I-O) pairs used to uniquely determine the initial condition from the given sequence $\{(\bar{u}_k, \bar{y}_k)\}_{k=0}^{L-1}$ in (3). In addition, N can be viewed as the prediction horizon in traditional MPC. Using collected I-O data, denoted by $(u^{\mathrm{d}}, y^{\mathrm{d}})$, we can decompose the Hankel matrices of (3) as follows:

$$\mathscr{H}_L(u^{\mathrm{d}}) = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \quad \mathscr{H}_L(y^{\mathrm{d}}) = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}, \tag{4}$$

where $U_p \in \mathbb{R}^{mT_{\text{ini}} \times (T-L+1)}$ and $Y_p \in \mathbb{R}^{pT_{\text{ini}} \times (T-L+1)}$ are the portions of the Hankel matrices used for estimating the initial condition (i.e., past), and $U_f \in \mathbb{R}^{mN \times (T-L+1)}$ and $Y_f \in \mathbb{R}^{pN \times (T-L+1)}$ are the portions used for prediction (i.e., future). A necessary condition for ensuring the information in the Hankel matrices is sufficiently rich is that T much be chosen such that $T \geq (m+1)(T_{\text{ini}} + N + n) - 1$.

III. Data-Driven Motion Planner

This section provides a brief overview of data-driven predictive control and outlines the application to trajectory planning for a quadruped. We further discuss similarities between the SRB template model and the data-driven model.

A. Data-Driven Predictive Control

This section outlines an approach to address predictive control without a physics-based model. In particular, we consider the DeePC methodology provided in [21], [22] as follows:

$$\begin{aligned} \min_{(g,u,y,\sigma)} \quad & \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\text{des}}\|_Q^2 + \|u_k\|_R^2 \right) + \lambda_g \|g\|^2 + \lambda_\sigma \|\sigma\|^2 \\ \text{s.t.} \quad & \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g + \begin{bmatrix} 0 \\ \sigma \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix} \\ & u_k \in \mathcal{U}, \quad y_k \in \mathcal{Y}, \quad k = 0, \ldots, N-1 \end{aligned} \tag{5}$$

where $Q \in \mathbb{R}^{p \times p}$ and $R \in \mathbb{R}^{m \times m}$ are positive definite weighting matrices, $\|y\|_Q^2 := y^\top Q\, y$, $\{y_k^{\text{des}}\}_{k=0}^{N-1}$ represents a desired trajectory, and $\mathcal{U}$ and $\mathcal{Y}$ are feasible sets. In addition, λ_g and λ_σ are positive weighting factors meant to regularize g and penalize the defect variable σ, respectively. Here, the defect variable σ allows (5) to remain feasible in the wake of noisy measurements. If no noise is present, then Theorem 1 applies directly. In our notation, $(u_{\text{ini}}, y_{\text{ini}})$ denotes the past measured trajectory (i.e., *feedback*) over the estimation horizon T_{ini} to be used to indirectly estimate the initial condition in (5). In addition, (u, y) represents the predicted trajectory over the control horizon N. We remark if the standard system identification approach is applied to compute the realization matrices in (1) optimally, the state vector may not correspond to a physically measurable variable. Hence, one would need to integrate the MPC approach with observer techniques to asymptotically estimate the states. However, the DeePC approach does not require any estimation beyond what was required during data collection.

While effective, the size of the optimization problem (5) is prohibitive for real-time implementation on a quadruped. Lengthening the prediction horizon by one results in an increase of $2(m+p)$ decision variables, and adds corresponding constraints. Furthermore, the majority of results in behavioral systems theory are applicable only to LTI systems. Extending these methods to nonlinear and underactuated dynamical models of legged robots requires larger sets of data (i.e., larger T). This introduces considerably more decision variables since the size of g is directly proportional to the size of T. For this reason, we consider a least-squares approximation of (5), which reduces the problem by $(p\,T_{\text{ini}} + T - L + 1)$ decision variables. In particular, a least-squares approximation is used to find g such that it can be removed from the problem, resulting in a constant linear mapping between the inputs u and the outputs y based solely on experimental data. We remark that using this approach with sufficiently large amounts of data precludes the need for σ in (5). Analogous to [22], finding an approximation of g reduces to the following offline optimization problem

$$\begin{aligned} \min_{g} \quad & \|g\|^2 \\ \text{s.t.} \quad & \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}. \end{aligned} \tag{6}$$

The closed-form expression of (6) can be described by

$$g = \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}, \tag{7}$$

where $(\cdot)^\dagger$ represents the pseudo inverse. Using the fact that $y = Y_f\, g$ from (5), we have

$$y = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}, \quad \mathcal{G} := Y_f \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger, \tag{8}$$

where $\mathcal{G}$ denotes the *data-driven state transition matrix over N-steps*. Using (8), we are now in position to present the general form of a computationally tractable predictive controller based on data for trajectory planning

$$\begin{aligned}\min_{(u,y)} \quad & \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\text{des}}\|_Q^2 + \|u_k\|_R^2 \right) \\ \text{s.t.} \quad & y = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} \\ & u_k \in \mathcal{U},\ y_k \in \mathcal{Y}, \quad k = 0, \dots, N-1. \end{aligned} \tag{9}$$

***Remark** 1:* Careful consideration is required when performing this approximation. In particular, we remark that g in (5) seeks to find a linear combination of the previous I-O pairs that can uniquely predict the future I-O pairs. The variables u and y are, in turn, directly determined by the choice of g and the data in the Hankel matrices. From Theorem 1, if properly constructed, all possible trajectories of (1) are in the range space of the Hankel matrices. However, this places no restriction on the norm of g. Suppose that we are interested in maintaining a constant non-zero velocity of a rigid body. In this case, position changes monotonically and $\|g\|_2 \to \infty$ as $t \to \infty$. Therefore, this restricts us to outputs that will remain in a neighborhood of zero.

B. Trajectory Planning for Quadrupedal Robots

In this section, we discuss the application of the data-driven predictive control of (9) to the real-time planning of quadrupeds and draw relations to the common SRB template model. The nonlinear SRB model is described by [9], [11], [12]

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} p_c \\ \dot{p}_c \\ R \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{p}_c \\ \frac{1}{m^{\text{net}}} f^{\text{net}} - g_0\, e_z \\ R\, \hat{\omega} \\ I_r^{-1}(R^\top \tau^{\text{net}} - \hat{\omega}\, I_r\, \omega) \end{bmatrix}, \tag{10}$$

where m^{net} is the total mass, g_0 is the gravitational constant, $e_z := \text{col}(0, 0, 1)$ is the unit vector along the z-axis, I_r is the body inertia, $p_c \in \mathbb{R}^3$ is the position of the COM of the robot in an inertial world frame, $\omega \in \mathbb{R}^3$ is the angular velocity in the body frame, $R \in \text{SO}(3)$ is the rotation matrix with respect to the inertial world frame, f^{net} is the net force acting on the COM, and τ^{net} is the net torque induced by the forces at the leg ends acting about the COM. Furthermore, we denote the skew symmetric operator by $\hat{(\cdot)} : \mathbb{R}^3 \to \mathfrak{so}(3)$. The net forces and torques in (10) can be described by

$$\begin{bmatrix} f^{\text{net}} \\ \tau^{\text{net}} \end{bmatrix} = \sum_{\ell \in \mathcal{C}} \begin{bmatrix} I_{3\times 3} \\ \hat{d}_\ell \end{bmatrix} f_\ell, \tag{11}$$

where $\ell \in \mathcal{C}$ represents the index of the contacting leg with the ground, $\mathcal{C}$ is the set of contacting points, $f_\ell \in \mathbb{R}^3$ is the ground reaction force (GRF) at leg ℓ, and d_ℓ is the vector from the COM to leg ℓ. The equations are nonlinear and typically linearized before being used with traditional MPC approaches. Due to the accuracy degradation over long prediction horizons induced by linearization and computational issues, the prediction horizon in these approaches is usually small. Since the horizon is small, many implementations for nominal gaits such as trotting assume the number of contact points with the environment remains constant for the duration of the MPC. However, multiple domains have also been considered for more dynamic gaits [32].

In the data-driven approach, we aim to draw on knowledge of the well-studied SRB model to pick suitable inputs and outputs while considering some of the potential pitfalls listed. In particular, the inputs and outputs used to construct the Hankel matrices are chosen to be $u := f \in \mathbb{R}^{12}$ (i.e., GRFs) and $y := \text{col}(\text{z}, \dot{\text{x}}, \dot{\text{y}}, \dot{\text{z}}, \alpha, \omega) \in \mathbb{R}^{10}$, where $\alpha \in \mathbb{R}^3$ denotes the Euler angles of the trunk. In other words, the inputs and outputs for the data-driven model are identical to those used in the SRB model (10), with the exception of the position in the transverse plane, i.e., the x and y position of the COM. These states are removed in light of Remark 1.

***Remark** 2:* Contrary to the SRB model, the data-driven model does not directly consider the mapping between the forces and the torques acting about the COM as in (11). It is assumed that the data-driven model encapsulates this mapping. While one could consider the relative foot positions directly in the model, the increase in the size makes this prohibitive for real-time computation.

The data-driven trajectory planner is then defined by

$$\begin{aligned}\min_{(u,y)} \quad & \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\text{des}}\|_Q^2 + \|u_k - u_k^{\text{des}}\|_R^2 \right) \\ \text{s.t.} \quad & y = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} \\ & u_k \in \mathcal{FC},\ y_k \in \mathcal{Y}, \quad k \in 0, \dots, N-1, \end{aligned} \tag{12}$$

where u_k^{des} represents the desired force at time $k \in \mathbb{Z}_{\geq 0}$ and $\mathcal{FC} := \{\text{col}(f_x, f_y, f_z) | f_z > 0,\ \pm f_x \leq \frac{\mu}{\sqrt{2}} f_z,\ \pm f_y \leq \frac{\mu}{\sqrt{2}} f_z\}$ denotes the linearized friction cone with μ being the friction coefficient. In order to address the fact that we are predicting over a larger horizon compared to many traditional SRB-based MPC approaches due to the lack of terminal cost, the desired force and the constraints on the forces should be considered carefully. In particular, the prediction horizon considered in this work is 1.25 times longer than the nominal stance time of 200 (ms), which guarantees the prediction will span multiple continuous-time domains (i.e., different stance leg configurations). Therefore, the desired force trajectory changes in a step-like manner at *anticipated* domain changes. The desired forces in the x, y, and z direction for leg ℓ are defined as $u_{k,\ell}^{\text{des}} := \text{col}(0, 0, \frac{m^{\text{net}} g_0}{N_{c,k}})$, $\forall \ell \in \mathcal{C}_k$ and zero otherwise. In this notation, $\mathcal{C}_k$ is the anticipated set of contacting legs with the ground at time k and $N_{c,k}$ represents the number of contacting legs at time k. The force constraints also change in a similar manner such that the forces on anticipated swing legs are restricted to zero, while the stance leg forces must abide by the linearized friction cone $\mathcal{FC}$. By altering the desired contact sequence, one could parameterize different gaits such as walking and trotting. Although $m^{\text{net}} g_0$ may not be strictly known, one could use the average net force obtained during the data collection procedure.

This data-driven predictive controller embodies many of the same principles as the SRB-based MPC. However, in the data-driven approach, we explicitly consider domain changes in

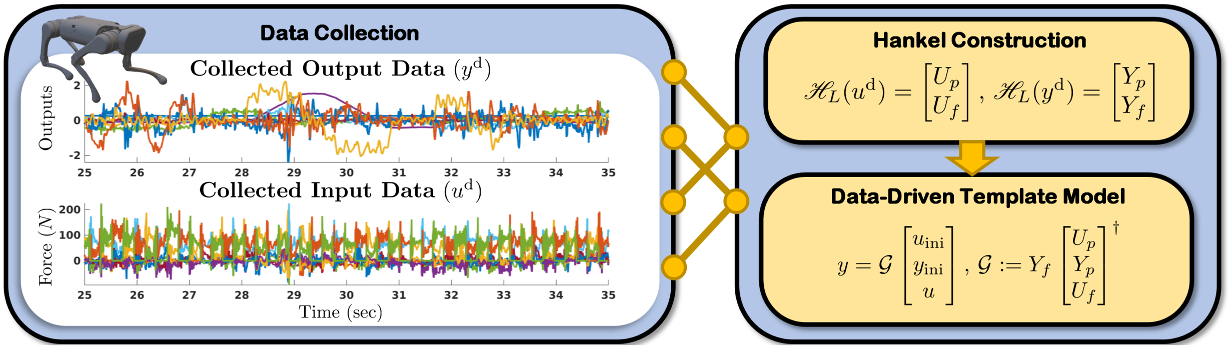
Data Collection
Collected Output Data (y^{d})
Outputs
2
0
-2
25 26 27 28 29 30 31 32 33 34 35
Collected Input Data (u^{d})
Force (N)
200
100
0
25 26 27 28 29 30 31 32 33 34 35
Time (sec)
Hankel Construction
$\mathscr{H}_L(u^{\mathrm{d}}) = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \mathscr{H}_L(y^{\mathrm{d}}) = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}$
Data-Driven Template Model
$y = \mathcal{G} \begin{bmatrix} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} \\ u \end{bmatrix}, \mathcal{G} := Y_f \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^{\dagger}$

Fig. 2. Overview of the process used to construct the data-driven template mode. The data is collected by directly using the QP-based low-level controller (17), and that data is then used to construct a template model on which a predictive trajectory planner can be based.

the prediction and do not consider the x and y position of the COM. In addition, no assumptions are made about the dynamics of the legs, enabling this approach to potentially capture some of the rich nonlinear dynamics indirectly through the GRF. Finally, this approach uses a constant mapping that does not require successive linearization as done in [9], [11], [12].

IV. Nonlinear Low-Level Controller

This section aims to present the low-level controller used to track the trajectories produced by the trajectory planner.

A. *Full-Order Nonlinear Dynamics*

Here we provide an overview of the full-order model used for the synthesis of the low-level controller. The model of the robot is constructed as a floating base, where $q \in \mathcal{Q} \subset \mathbb{R}^{n_q}$ represents the generalized coordinates, $\mathcal{Q}$ is the configuration space, and n_q denotes the number of DOFs. We then define the state vector to be $x_f := \mathrm{col}(q, \dot{q}) \in \mathcal{X} \subset \mathbb{R}^{2n_q}$ with $\mathcal{X} := \mathcal{Q} \times \mathbb{R}^{n_q}$. The motor torques are then described by $\tau \in \mathcal{T} \subset \mathbb{R}^{m_\tau}$ where $\mathcal{T}$ is the set of admissible torques and m_τ is the number of inputs. The equations of motions are described by

$$D(q)\,\ddot{q} + H(q,\dot{q}) = \Upsilon\,\tau + J^\top(q)\,f, \tag{13}$$

where $D(q) \in \mathbb{R}^{n_q \times n_q}$ represents the mass-inertia matrix, $H(q,\dot{q}) \in \mathbb{R}^{n_q}$ denotes the Coriolis, centrifugal, and gravitational terms, $\Upsilon \in \mathbb{R}^{n_q \times m_\tau}$ represents the input matrix, $J(q)$ denotes the contact Jacobian matrix, and $f := \mathrm{col}\{f_\ell | \, \ell \in \mathcal{C}\}$ represents the vector GRFs of the contacting leg ends. We further impose the holonomic constraint $\ddot{r} = 0$ on (13), where $r := \mathrm{col}\{p_\ell | \, \ell \in \mathcal{C}\}$ represents the position of the contacting leg ends with the environment. This constraint implies rigid contact with the ground and is valid if $f_\ell \in \mathcal{FC}, \forall \ell \in \mathcal{C}$.

B. *Virtual Constraints Controller*

This section provides the formulation of a QP-based virtual constraints controller used for tracking both the forces and COM trajectory provided by the trajectory planner. We consider a set of holonomic *virtual constraints* [31] as

$$h(x_f, t) := h_0(q) - h^{\mathrm{des}}(t), \tag{14}$$

where $h(x_f, t) \in \mathbb{R}^{n_{vc}}$, with n_{vc} representing the number of virtual constraints, and is imposed by I-O linearization [33]. The term $h_0(q)$ denotes the variables that we are interested in controlling, and $h^{\mathrm{des}}(t)$ describes the desired evolution of $h_0(q)$. In this work, $h_0(q)$ consists of the position and orientation of the COM, and the Cartesian position of the swing feet. In particular, a Bézier polynomial is constructed to move the foot from its initial position to the target position, wherein the target position is determined using the Raibert heuristic [34, Eq. (4), pp. 46]. Differentiating $h(x_f, t)$ twice along the dynamics (13), we have

$$\ddot{h} = \Theta_1(x_f)\,\tau + \Theta_2(x_f)\,f + \theta(x_f) = -K_P\,h - K_D\,\dot{h} + \delta, \tag{15}$$

where Θ_1, Θ_2, and θ are of proper dimension and obtained using a standard I-O linearization procedure. We refer the reader to [35, Appendix A.2] for more details on the derivation of these terms. In addition, K_P and K_D are positive definite gain matrices, and $\delta \in \mathbb{R}^{n_{vc}}$ is a defect variable used in the formulation of the QP. In a similar manner, we define the holonomic constraint placed on the stance legs to enforce rigid contact by differentiating the Cartesian coordinates at the stance leg ends twice and setting them to zero as follows:

$$\ddot{r} = \Phi_1(x_f)\,\tau + \Phi_2(x_f)\,f + \phi(x_f) = 0, \tag{16}$$

for some proper Φ_1, Φ_2, and ϕ. We are now in a position to present the QP-based nonlinear controller. The goal is to solve for the minimum 2-norm torques while imposing the virtual constraints and tracking the desired forces, as well as abiding by the feasible torques and friction cone. To this end, the following strictly convex QP is employed [36]

$$\begin{aligned} \min_{(\tau,f,\delta)} \; & \frac{\gamma_1}{2}\|\tau\|^2 + \frac{\gamma_2}{2}\|f - f^{\mathrm{des}}\|^2 + \frac{\gamma_3}{2}\|\delta\|^2 \\ \text{s.t.} \quad & \Theta_1(x_f)\,\tau + \Theta_2(x_f)\,f + \theta(x_f) = -K_P\,h - K_D\,\dot{h} + \delta \\ & \Phi_1(x_f)\,\tau + \Phi_2(x_f)\,f + \phi(x_f) = 0 \\ & \tau \in \mathcal{T}, \quad f_\ell \in \mathcal{FC}, \quad \forall \ell \in \mathcal{C}, \end{aligned} \tag{17}$$

where γ_1, γ_2, and γ_3 are positive weighting factors. In addition, the desired force profile $f^{\mathrm{des}}(t)$ represents the optimal GRFs (i.e., inputs u) prescribed by the high-level data-driven planner in (12). The defect variable δ is included such that the QP remains feasible if the I-O linearization cannot be met exactly. The weighting factor on δ is chosen to be much larger than the other weights to make the defect variable as small as possible. The low-level controller can be used without a planner if the virtual constraints are chosen heuristically, i.e., $h^{\mathrm{des}}(t)$ can be hand-tuned to produce stable locomotion.

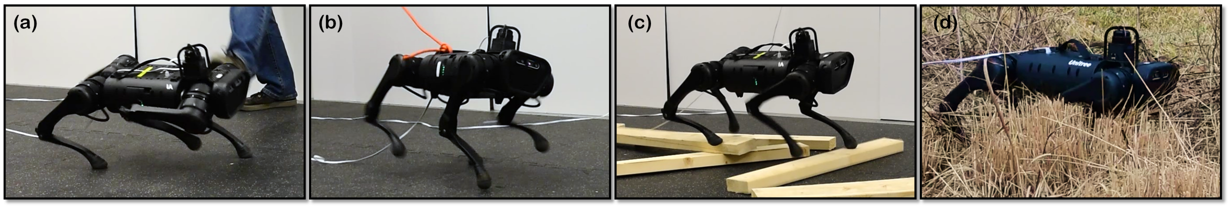

Fig. 3. Snapshots from experiments with the proposed hierarchical control algorithm: (a) external push disturbances, (b) external tethered pull disturbances, (c) unknown rough terrain covered with wooden blocks, and (d) unstructured and unknown outdoor environment.

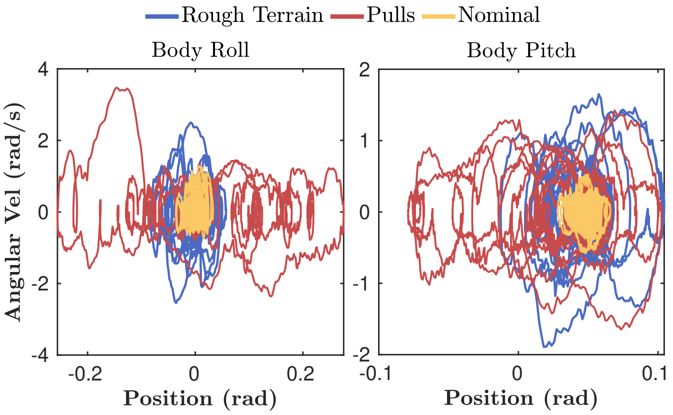

Fig. 4. Phase portraits of the robot's body orientation (i.e., roll and pitch) during different experiments. The quadruped is able to robustly trot over flat ground (nominal), unknown rough terrain covered with wooden blocks, and subject to external disturbances (pulls). For each experiment, the robot is commanded to walk forward at 0.5 (m/s). The reason for the slight pitch offset is unknown, but is attributed to tracking error at the low-level.

However, we aim to provide an optimal trajectory produced by a trajectory planner to reduce the required expertise necessary to enable stable locomotion.

V. Experimental Results

This section seeks to demonstrate the efficacy of the proposed approach for quadrupedal locomotion through a variety of hardware experiments. We consider the quadrupedal platform A1 made by Unitree. This robot consists of $n_q = 18$ DOFs. We consider a floating-base model of the robot, wherein the absolute position and orientation of the floating base comprise the first 6 DOFs, which are unactuated. The remaining DOFs are composed of the actuated leg joints. Each leg has a 2-DOF hip joint followed by a 1-DOF knee joint (i.e., $m_\tau = 12$). The robot weighs approximately 12.45 (kg) and stands roughly 28 (cm) off the ground.

A. Data Collection and Trajectory Planner

This section describes the procedure and parameters used for constructing the data-driven model. An overview of this procedure can be found in Fig. 2. The data for the Hankel matrices were collected at 100 (Hz) by moving the robot around a lab environment using a trot gait, commanded via a joystick, utilizing only the low-level controller presented in Section IV-B. From the low-level QP (17), we obtain estimates of the GRFs and these estimates are then utilized during the construction of the data-driven model as inputs u^{d}. Although we consider the use of the controller presented in Section IV-B, a different low-level controller can be used as long as the outputs can be properly estimated. As mentioned in Section III-B, the proposed outputs are taken as $y^{\mathrm{d}} = \mathrm{col}(\mathrm{z}, \dot{\mathrm{x}}, \dot{\mathrm{y}}, \dot{\mathrm{z}}, \alpha, \omega) \in \mathbb{R}^{10}$. We opt to use a joystick as opposed to a random input trajectory which may require more data due to the requirement of persistency of excitation but does not pose an issue in the current formulation due to the removal of g from the predictive controller. Namely, the size of the high-level QP remains constant, regardless of the amount of data used. The parameters used are $T_{\mathrm{ini}} = 10$ for the estimation horizon, $N = 25$ for the prediction horizon, and $T = 4284$ collected I-O data points, which is much greater than the minimum number of data points required by the general theory. The use of a large amount of data is highly beneficial here because the system is nonlinear. By using more data, the model better encapsulates information from various configurations and is less sensitive to noise from the collected data, providing a better approximation of the system. This is inline with the promising results of [21], [22] for control of nonlinear systems. In particular, [21] considers drone dynamics that are similar to the SRB model.

Although the size of the problem is reduced considerably by using (9) as opposed to (5), it is still large with 550 decision variables and 800 constraints. The planner is solved using OSQP [37] and takes upwards of approximately 25 (ms) to solve on an external laptop with an Intel® Core™ i7-1185G7 running at 3.00 GHz and 16 GB of RAM. We therefore run the planner every 30 (ms) and use the first three steps of the predicted COM trajectory and GRFs as inputs passed to the low-level controller. Finally, the parameters in the predictive controller are taken to be $Q = \mathrm{diag}(8e6, 5e5, 5e5, 5e3, 8e6, 8e5, 8e5, 5e3, 5e3, 5e5)$ and $R = 0.5I$, where I is the identity of appropriate size.

***Remark** 3:* If good force estimates are not available, the chosen I-O pair seems restrictive. To alleviate this, one could also consider using $u := \mathrm{col}(\mathrm{z}^{\mathrm{des}}, \dot{\mathrm{x}}^{\mathrm{des}}, \dot{\mathrm{y}}^{\mathrm{des}}, \dot{\mathrm{z}}^{\mathrm{des}}, \alpha^{\mathrm{des}}, \omega^{\mathrm{des}})$, $y := \mathrm{col}(\mathrm{z}, \dot{\mathrm{x}}, \dot{\mathrm{y}}, \dot{\mathrm{z}}, \alpha, \omega)$ as the I-O pair for (12), which is less restrictive in terms of readily available measurements.

B. Data-Driven Experimental Results

The purpose of this section is to provide the parameters for the QP-based low-level controller (17) used in tandem with the trajectory planner and further provide experimental results of the proposed hierarchical control scheme. In order to track the provided trajectory, the weights in the low-level QP are chosen to be $\gamma_1 = 10^2$, $\gamma_2 = 10^3$, and $\gamma_3 = 10^6$. The low-level controller is solved at 1kHz using qpSWIFT [38] and takes approximately 0.22 (ms) using the same external laptop as the planner. Snapshots of various experiments using

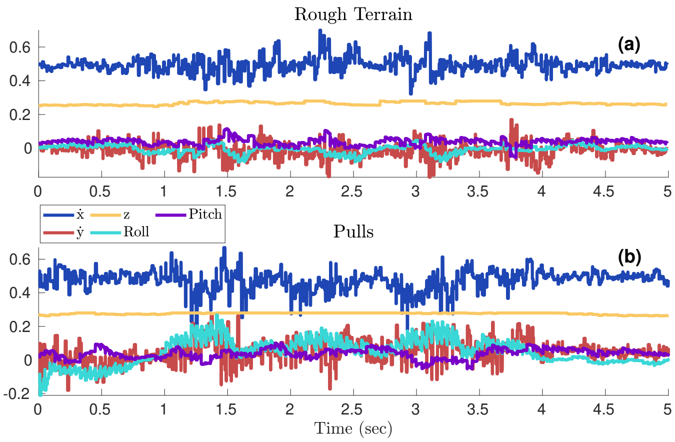

Fig. 5. The prescribed trajectory from the planner while trotting subject to (a) rough terrain consisting of unstructured wooden blocks and (b) tethered pulls. The robot is commanded to walk forward at 0.5 (m/s), the height command is 0.28 (m), and all other states are commanded to be zero. Pulls occur for the first 4 seconds.

the trajectory planner in tandem with the low-level controller can be found in Fig. 3. In these experiments, the robot is commanded to blindly walk forward at 0.5 (m/s) and was subject to pushes (Fig. 3(a)), pulls (Fig. 3(b)), unknown rough terrain (Fig. 3(c)), and unstructured outdoor environments (Fig. 3(d)). In all scenarios, the quadruped was able to robustly maneuver. Videos of the experiments can be found online at [39]. Phase portraits for these stable gaits can be found in Fig. 4. The phase portraits remain small and bounded, which demonstrates the overall stability of the system. Using the data from the same experiments found in Fig. 4, Fig. 5 displays the time response of the trajectories resulting from the planner. While the disturbances are unknown, the planner remains stable showing the robustness of the planner against unknown external influences.

Extension to Other Gaits: The controller was additionally evaluated in terms of its ability to track a time-varying reference, and to consider an additional gait without collecting new data. In order to test this, the robot was maneuvered across flat ground using a joystick for velocity commands. The comparison between the output of the planner and the commanded velocities for a trot gait can be found in Fig. 6(a), and for a walk gait in Fig. 6(b). Additional experiments also evaluated the efficacy of the planner when using a stance time that is 25% shorter (150 (ms)) and longer (250 (ms)) than that which was used during the initial data collection. While omitted due to space constraints, the videos of these experiments can be found online at [39]. Our results suggest that the same data can be used even in situations that were not exactly represented during the data collection procedure. This includes being robust to external disturbances and handling gaits with different footfall patterns and step frequencies than that which was used during collection. However, dynamic gaits like bounding may require additional data collection.

C. Comparison to Physics-Based Reduced-Order Model

This section aims to briefly provide insight into how the proposed data-driven methodology compares to linearized SRB. A comparison of the trajectories of the proposed approach versus the linearized SRB can be found in Fig. 7. The proposed approach, using only data to construct a model, performs comparably to a moderately tuned linearized SRB-based MPC. The slightly attenuated noise profile in the proposed approach is likely due to the estimation that is inherently contained within the model through $(u_{\text{ini}}, y_{\text{ini}})$. This could also be due to the longer time horizon of the proposed approach and the fact that the horizon spans multiple domains. While the two methods perform very similarly, the primary advantage of the proposed approach is that no knowledge of the system dynamics is required to create a reduced-order model and foregoes the need for explicit system identification. Improvements could potentially be obtained by considering a Page matrix representation [23] or singular value truncation [40], but we leave this to future investigation. Videos of the comparison can be found in [39].

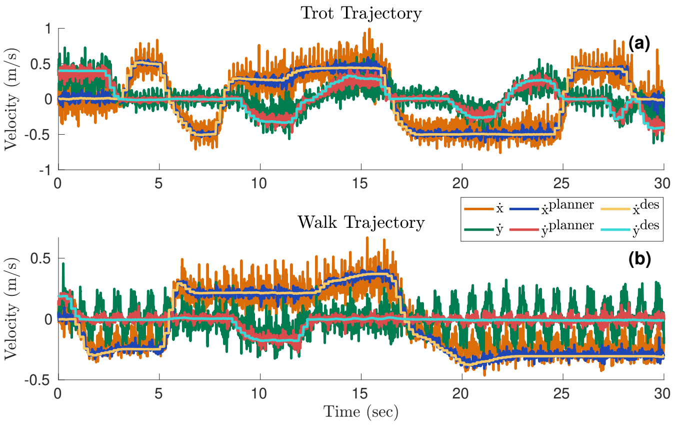

Fig. 6. The figure shows the stable output tracking of the planner compared to the time-varying reference provided by a user through a joystick and the robot's actual states while using (a) a trot gait and (b) a walk gait. Each domain lasts 200 (ms).

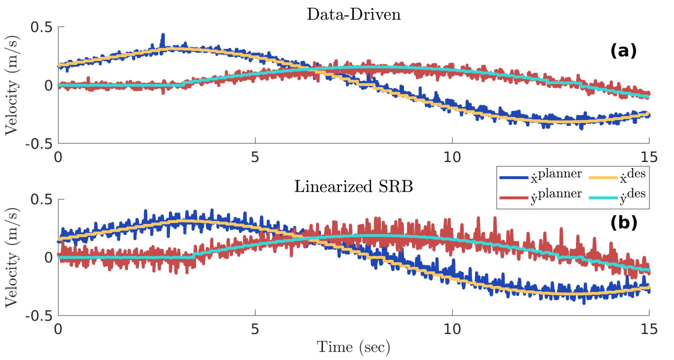

Fig. 7. Hardware experiments showing the evolution of the trajectory produced by the data-driven planner (a) and a MPC planner using a linearized SRB model (b). The robot aims to follow a velocity profile that results in a circular path.

VI. Conclusion

This paper presented a hierarchical control algorithm based on data-driven template models for real-time planning and control of dynamic quadrupedal robots. At the higher level, we provide a reduced-order model, based purely on data, which is used in a computationally tractable predictive control framework for real-time trajectory planning. The data-driven model leverages the information about the SRB model while forgoing the need for successive linearization. The optimal trajectories are then passed to a QP-based and low-level nonlinear controller for whole-body motion control. The efficacy of the proposed layered control approach is validated via extensive experiments for robustly stable locomotion of the A1 quadrupedal robot on different unknown terrains, in the presence of disturbances, and considering different gaits and gait parameters without collecting additional data. Future work should explore the use of data-driven template models with more complex systems such as collaborative systems. In particular, the scalability to large-scale complex systems will be a major challenge. Exploring how the data-driven approach compares analytically to the linearized SRB model would also provide valuable insight into the dynamics captured by the proposed method.

References

[1] R. J. Full and D. E. Koditschek, "Templates and anchors: Neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, pp. 3325–3332, December 1999.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation*, vol. 2, Sep. 2003, pp. 1620–1626.

[3] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2017, pp. 667–673.

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

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

[6] 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.

[7] P. Holmes, R. J. Full, D. E. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: Models, analyses, and challenges," *SIAM Review*, vol. 48, no. 2, pp. 207–304, May 2006.

[8] A. Hereid, S. Kolathaya, M. S. Jones, J. Van Why, J. W. Hurst, and A. D. Ames, "Dynamic multi-domain bipedal walking with ATRIAS through SLIP based human-inspired control," in *International Conference on Hybrid Systems: Computation and Control*. ACM, 2014, pp. 263–272.

[9] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2018, pp. 2245–2252.

[10] 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.

[11] 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.

[12] 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.

[13] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2003, pp. 1644–1650.

[14] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1550039, 2016.

[15] 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*, 2014, pp. 295–302.

[16] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini, "MPC-based controller with terrain insight for dynamic legged locomotion," in *IEEE International Conference on Robotics and Automation*. IEEE, 2020, pp. 2436–2442.

[17] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

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

[19] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, 2022.

[20] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2020.

[21] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *European Control Conference*, 2019, pp. 307–312.

[22] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for grid-connected power converters," in *IEEE Conference on Decision and Control*, 2019, pp. 8130–8135.

[23] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, 2021.

[24] H. J. Van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4753–4768, 2020.

[25] L. Wei, Y. Yan, and J. Bao, "A data-driven predictive control structure in the behavioral framework," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 152–157, 2020.

[26] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[27] J. C. Willems, "From time series to linear system—Part I. Finite dimensional linear time invariant systems," *Automatica*, vol. 22, no. 5, pp. 561–580, 1986.

[28] I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor, *Exact and approximate modeling of linear systems: A behavioral approach*. SIAM, 2006.

[29] J. Berberich and F. Allgöwer, "A trajectory-based framework for data-driven system analysis and control," in *European Control Conference*, 2020, pp. 1365–1370.

[30] C. Verhoek, H. S. Abbas, R. Tóth, and S. Haesaert, "Data-driven predictive control for linear parameter-varying systems," *IFAC-PapersOnLine*, vol. 54, no. 8, pp. 101–108, 2021.

[31] E. Westervelt, J. Grizzle, C. Chevallereau, J. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Taylor & Francis/CRC, 2007.

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

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

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

[35] J. Kim and K. Akbari Hamed, "Cooperative locomotion via supervisory predictive control and distributed nonlinear controllers," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 3, Dec 2021.

[36] R. T. Fawcett, A. Pandala, A. D. Ames, and K. A. Hamed, "Robust stabilization of periodic gaits for quadrupedal locomotion via qp-based virtual constraint controllers," *IEEE Control Systems Letters*, vol. 6, pp. 1736–1741, 2021.

[37] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[38] 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.

[39] Toward a data driven template model for quadrupedal locomotion. [Online]. Available: https://youtu.be/VJ54siiF0lc.

[40] S. Chatterjee, "Matrix estimation by universal singular value thresholding," *The Annals of Statistics*, vol. 43, no. 1, pp. 177–214, 2015.