Distributed Data-Driven Predictive Control for Multi-Agent Collaborative Legged Locomotion

Randall T. Fawcett[1], Leila Amanzadeh[1], Jeeseop Kim[1], Aaron D. Ames[2], and Kaveh Akbari Hamed[1]

***Abstract*— The aim of this work is to define a planner that enables robust legged locomotion for complex multi-agent systems consisting of several holonomically constrained quadrupeds. To this end, we employ a methodology based on behavioral systems theory to model the sophisticated and high-dimensional structure induced by the holonomic constraints. The resulting model is then used in tandem with distributed control techniques such that the computational burden is shared across agents while the coupling between agents is preserved. Finally, this distributed model is framed in the context of a predictive controller, resulting in a robustly stable method for trajectory planning. This methodology is tested in simulation with up to five agents and is further experimentally validated on three A1 quadrupedal robots subject to various uncertainties, including payloads, rough terrain, and push disturbances.**

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

I. Introduction

This work investigates multi-agent systems composed of high-dimensional quadrupedal robots that are rigidly holonomically constrained to one another using ball joints, introducing high interaction forces (see Fig. 1). Current state-of-the-art approaches, even when considering only a single agent, generally involve using a reduced-order model of some kind [1]. For a single agent, there have been many template models that have worked effectively, as will be discussed shortly. However, template models for multi-agent systems, particularly those with large interaction forces, have not yet been fully explored. In particular, it is difficult to use traditional modeling techniques to model an increasing number of agents due to increased dynamic complexity. Even in the case that a large-scale dynamical system with strong interaction forces could be modeled at the reduced-order level, it would likely be of such complexity that it would no longer *efficiently* function as a reduced-order model for control. The goal of this work, therefore, is to address these issues. Namely, we aim to synthesize data-driven template models for planning of large multi-agent systems such that the resulting planner is computationally efficient for use in real time.

The work of R. T. Fawcett is supported by the National Science Foundation (NSF) under Grant 2128948. The work of K. Akbari Hamed is supported by the NSF under Grants 1924617 and 2128948. The work of A. D. Ames is supported by the NSF under Grant 1924526.

[1]R. T. Fawcett, L. Amanzadeh, J. Kim, and K. Akbari Hamed (Corresponding Author) are with the Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA, 24061, USA, {randallf, leila7, jeeseop, kavehakbarihamed}@vt.edu

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

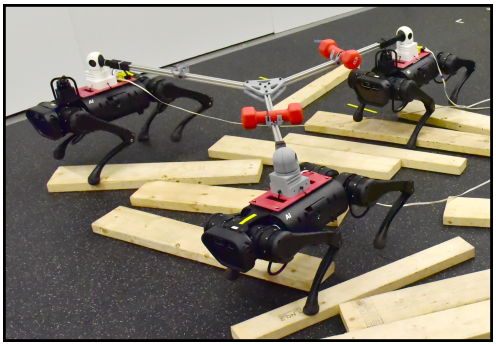

Fig. 1. Snapshot showing the locomotion of three holonomically constrained A1 robots on wooden blocks.

A. Multi-Agent Systems

Multi-Agent systems have been a major area of research, particularly in the context of cooperative manipulation [2], [3], control of unmanned aerial vehicles [4], [5], and ground vehicles [6], [7]. However, when considering many agents, the system is usually high-dimensional, which also motivates distributed approaches [8]–[10]. These methods reduce the computational burden for controllers and planners, but they have also *not* been readily extended to legged robotic systems with high dimensionality, unilateral constraints, and high interaction forces, as is the case in this work. Planning for multi-agent systems also usually necessitates the use of a reduced-order model [11], but in the case of holonomically constrained quadrupeds, the best model is unclear.

B. Reduced-Order Models

In order to create predictive controllers for complex systems, it is often necessary to utilize a reduced-order model. In terms of legged locomotion, a very well-known and studied model is that of the linear inverted pendulum (LIP) [12], which has been used for both quadrupedal and bipedal locomotion. However, the LIP model generally enforces quasi-static motions due to the requirement that the center of pressure remains within the convex hull formed by the contacting points with the environment. Nonetheless, the model's simplicity has been of great interest over the years and has been used extensively in both simulation and experiments using various platforms [12]–[16]. It is also worth mentioning that this model is not amenable to adding moments induced about the center of mass (COM) by external forces [17], [18], which is important when dealing with holonomically constrained multi-agent systems.

More recently, the single rigid body (SRB) model has gained traction to achieve dynamic locomotion [19]. This reduced-order model has proven to be effective in creating

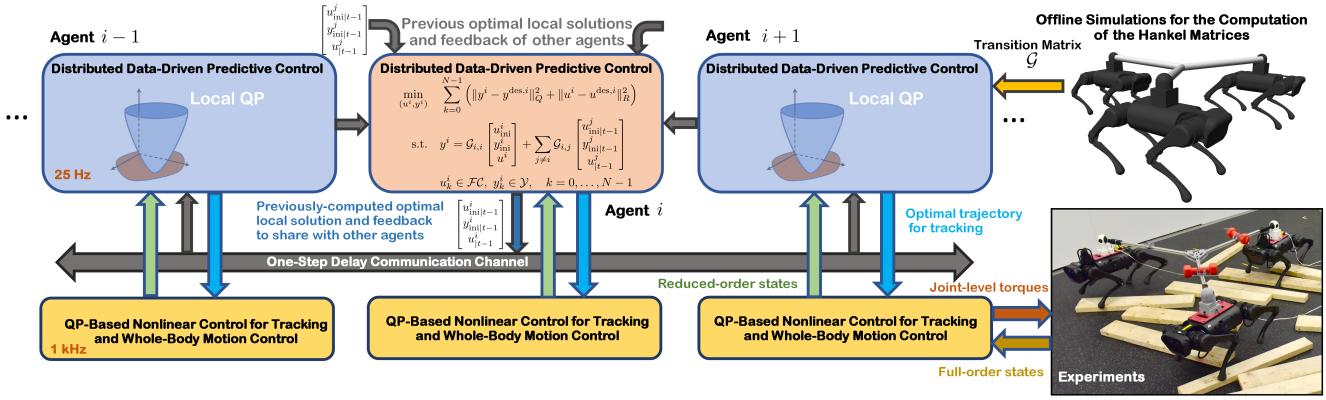

Fig. 2. Overview of the proposed control algorithm with distributed DDPC algorithms at the high level for trajectory optimization of cooperative locomotion and nonlinear controllers at the low level for tracking and whole-body motion control.

natural and dynamic motions for various legged robot platforms [19]–[22]. However, a drawback is that the legs are assumed to be massless for this model, which can become difficult when using robots with more mass. Conversely, one major benefit of the SRB model, in contrast to the LIP model, is that it allows one to consider the moments induced about the COM caused by external forces, which is useful in this case because of the holonomic constraints. Unfortunately, using the SRB model subject to interaction forces among holonomically constrained agents becomes prohibitively complex as the number of agents increases, which motivates the use of data-driven methods.

C. Data-Driven Methods

Data-driven methods are becoming more popular as systems become more complex [23]. This motivates their use in multi-agent systems, especially in the case where the individual agents are complex. Even though individual systems can be modeled adequately, their combined dynamics make the system increasingly sophisticated, and an effective reduced-order model has yet to be determined. One such method recently popularized in the robotics community is reinforcement learning [24], [25]. This method has been used for collaborative locomotion among multiple quadrupeds [26] but takes considerable time and computation power. Furthermore, reinforcement learning provides little intuition as to the inner workings of the result. In this work, we pursue a different avenue.

Here, we consider the use of data-driven methods in the context of behavioral systems theory. This methodology parameterizes a linear time-invariant (LTI) system directly in terms of its measured trajectories [27]–[29]. Behavioral systems theory, when used in a predictive control or planning framework, is usually referred to as data-driven predictive control (DDPC) or data-enabled predictive control (DeePC) and has lately become of great interest to the robotics community [30]–[36]. Recently, extensions have been made to use such methods for certain types of nonlinear systems [37] and linear parameter varying systems [38], but there have yet to be any proofs extending to a broad class of general nonlinear systems. There have also been advances in using these methods for stochastic implementations, and in particular, there have been experimental validations for nonlinear systems, even though the theory does not directly apply [31]–[33], [36]. However, to the best of the authors' knowledge, there have not been implementations for multi-agent systems. We further note that the extension to collaborative legged locomotion introduces many difficulties, including the hybrid nature of legged locomotion and unilateral constraints.

D. Goals, Objectives, and Contributions

The *overarching goal* of this paper is to develop a computationally tractable real-time predictive planner for collaborative legged locomotion using a behavioral approach. Namely, this work's *objectives* and *key contributions* are enumerated as follows: 1) A model is created using concepts from behavioral systems theory for systems of quadrupeds that are holonomically constrained to one another. 2) The model is used in the context of a *distributed* predictive control framework such that each agent can effectively plan for its own motions while considering the motions of other agents (see Fig. 2). 3) Simulation results for 5 constrained quadrupeds in the presence of uncertainty are provided. 4) We present extensive experimental validation on a team of 3 holonomically constrained quadrupeds. The experimental validation shows robust locomotion of the A1 robot subject to various uncertainties, including rough terrain, push disturbances, and outdoor environments.

II. Preliminaries

In this section, we overview some important concepts from behavioral systems theory that will be used throughout this work. Behavioral systems theory provides a manner in which data collected from a system can be leveraged to directly create a model. In particular, consider an LTI model 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 $k \in \mathbb{Z}_{\geq 0} := \{0, 1, \cdots\}$. Such a model can be represented in discrete time as follows:

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

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. In this notation, n, m, and p represent the number of states, inputs, and outputs, respectively. One difference between traditional system identification and behavioral systems theory is that,

traditionally, one would attempt to reconstruct the matrices of (1) using data. In the behavioral context, we obtain an input-output (I-O) model without reconstructing the state matrices.

To introduce the concepts, consider $L, T \in \mathbb{Z}_{\geq 0}$, where $T \geq L$. In addition, define some input trajectory $u^{\mathrm{d}} \in \mathbb{R}^{mT}$ composed of a sequence of u_k^{d}, i.e., $u^{\mathrm{d}} := \mathrm{col}(u_0^{\mathrm{d}}, \ldots, u_{T-1}^{\mathrm{d}})$. In this notation, "col" represents the column operator. Using this trajectory, one can construct the following Hankel matrix

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

Definition 1 ([30]): The signal u^{d} is said to be *persistently exciting* of order L if $\mathscr{H}_L(u^{\mathrm{d}})$ is full row rank.

Definition 2 ([30]): The sequence $\{(u_k, y_k)\}_{k=0}^{T-1}$ 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 can now present a foundational theorem in behavioral systems theory, used to represent an LTI system based on observed trajectories.

***Theorem** 1:* [27, Theorem 1] *Let an observed trajectory of* (1)*, 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 provides a constructive manner in which an LTI system can be represented through its trajectories without system identification. This concept will be used throughout this work as we aim to parameterize a complex system of robots by using only their trajectories. To do so, we consider two different horizons denoted by T_{ini} and N, which represent the estimation horizon and control horizon, respectively. The estimation horizon encapsulates the input-output pairs that are required in order to determine the initial conditions of a trajectory given a particular I-O sequence $\{(\bar{u}_k, \bar{y}_k)\}_{k=0}^{L-1}$ from (3). In contrast, the prediction horizon represents how far into the future predictions of the trajectories are made, similar to that of traditional model predictive control (MPC). Finally, we define $L = T_{\mathrm{ini}} + N$ for compact notation. We denote the collected I-O data by $(u^{\mathrm{d}}, y^{\mathrm{d}})$ and can decompose the Hankel matrices of (3) into two parts 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 $U_p \in \mathbb{R}^{mT_{\mathrm{ini}} \times (T-L+1)}$ and $Y_p \in \mathbb{R}^{pT_{\mathrm{ini}} \times (T-L+1)}$ 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 and sufficient condition to establish that the Hankel matrix is sufficiently rich is to choose T such that $T \geq (m+1)(T_{\mathrm{ini}}+N+n)-1$, which is a well-known result in behavioral systems theory.

III. Distributed DDPC for Trajectory Planning

In this section, we present our main contribution—the development of a DDPC for constrained multi-agent systems.

A. Data-Driven Predictive Control

The aim of this section is to outline how the data-driven approach is used for predictive control, as well as highlight some difficulties. To begin, consider the real-time DeePC methodology provided in [31], [32] as follows:

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

where $Q \in \mathbb{R}^{p \times p}$ and $R \in \mathbb{R}^{m \times m}$ are positive definite weighting matrices, $\|y\|_Q^2 := y^\top Q\, y$, $\{y_k^{\mathrm{des}}\}_{k=0}^{N-1}$ represents a desired trajectory, and $\mathcal{U}$ and $\mathcal{Y}$ are the feasible input and output sets, respectively. In our notation, $(u_{\mathrm{ini}}, y_{\mathrm{ini}})$ denotes the past measured trajectory over the estimation horizon T_{ini}, which provides feedback directly into the model. In addition, (u, y) represents the predicted I-O trajectory over the control horizon N. This method has proven to be robust and has worked for several nonlinear systems [31], [32]. One of the primary reasons for this is the addition of λ_g and λ_σ, which are positive weighting factors meant to regularize the g vector from Theorem 1 and penalize the defect variable σ, respectively. Note that σ is added to lessen the effect that noisy data has on the system. If the data were to contain no noise, this variable could be removed, though it is generally required in practice.

Although this methodology has the benefit of not requiring a direct model, it also comes with considerable computational complexity as the system increases in size due primarily to the vector g. Consequently, this method is intractable for real-time computation on teams of quadrupedal robots. For a more in-depth discussion on the matter, we refer the interested reader to [36]. To circumvent the problem, we adopt an offline approximation for g, analogous to [32], [36] as follows:

$$g = \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger \begin{bmatrix} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} \\ u \end{bmatrix}, \; y = \mathcal{G} \begin{bmatrix} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} \\ u \end{bmatrix}, \; \mathcal{G} := Y_f \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger, \tag{6}$$

where $(\cdot)^\dagger$ represents the pseudo inverse and $\mathcal{G}$ denotes the *data-driven state transition matrix over N-steps*. Using this procedure, we can remove g from the optimization problem (5), considerably reducing the number of decision variables.

B. Distributed Multi-Agent Trajectory Planning

Here, we outline how the data-driven model (6) can be used to create a distributed planner for groups of holonomically constrained legged robots. First, we present the control law in a centralized manner, i.e., assuming that

one planner is used to control the whole system. This is later decomposed for distributed computation. In particular, consider the centralized predictive control problem utilizing (6) as follows:

$$\begin{aligned}\min_{(u,y)} \quad & \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\text{des}}\|_Q^2 + \|u_k\|_R^2 \right) \\ \text{s.t.} \quad & y = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} \\ & u_k \in \mathcal{U},\ y_k \in \mathcal{Y}, \qquad k = 0, \dots, N-1. \end{aligned} \tag{7}$$

This method has been shown to be amenable to trajectory planning for single-agent legged robots [36] but has not been used for multi-agent systems. In the multi-agent context, the state transition matrix $\mathcal{G}$ is created using data from all of the agents. In particular, Theorem 1 considers the trajectory $(u^{\text{d}}, y^{\text{d}})$ to define the Hankel matrix. To construct the Hankel martix for multi-agent systems, we can define $u^{\text{d}} := \{\text{col}(u_k^{\text{d},1}, u_k^{\text{d},2}, \cdots, u_k^{\text{d},n_a})\}_{k=0}^{T-1}$ and $y^{\text{d}} := \{\text{col}(y_k^{\text{d},1}, y_k^{\text{d},2}, \cdots, y_k^{\text{d},n_a})\}_{k=0}^{T-1}$, where $(\cdot)_k^{\text{d},i}$ for all $i \in \mathcal{I} := \{1, \cdots, n_a\}$ denotes the data contributed by agent i at the sample time k, and n_a is the total number of agents. Using these combined I-O pairs, we obtain a large Hankel matrix describing the entire complex system. Furthermore, this new Hankel matrix can be decomposed according to (4), and the corresponding g vector can be approximated using (6).

In moving to multi-agent systems, however, it is desirable to share the computational load between agents. In order to do so, we consider a decomposition of $\mathcal{G}$ as follows:

$$\mathcal{G} = \begin{bmatrix} \mathcal{G}_{1,1} & \mathcal{G}_{1,2} & \cdots & \mathcal{G}_{1,n_a} \\ \mathcal{G}_{2,1} & \mathcal{G}_{2,2} & \cdots & \mathcal{G}_{2,n_a} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{G}_{n_a,1} & \mathcal{G}_{n_a,2} & \cdots & \mathcal{G}_{n_a,n_a} \end{bmatrix}, \tag{8}$$

where $\mathcal{G}_{i,j}$ represents the effect of agent j on the predicted output of agent i (i.e., coupling). In particular, the predicted output of agent $i \in \mathcal{I}$ can be expressed by

$$y^i = \mathcal{G}_{i,i} \begin{bmatrix} u_{\text{ini}}^i \\ y_{\text{ini}}^i \\ u^i \end{bmatrix} + \sum_{j \neq i} \mathcal{G}_{i,j} \begin{bmatrix} u_{\text{ini}}^j \\ y_{\text{ini}}^j \\ u^j \end{bmatrix}, \tag{9}$$

where $(\cdot)^i$ represents the variables related to the ith agent. To alleviate the coupling problem, we adopt a *one-step communication delay protocol*. In particular, we assume that at every time sample t, each local DDPC has access to the optimal predicted solutions of the other local DDPCs and their past measurements at time $t-1$. Using this assumption, the predicted output of agent i can be approximated by

$$y^i \approx \mathcal{G}_{i,i} \begin{bmatrix} u_{\text{ini}}^i \\ y_{\text{ini}}^i \\ u^i \end{bmatrix} + \sum_{j \neq i} \mathcal{G}_{i,j} \begin{bmatrix} u_{\text{ini}|t-1}^j \\ y_{\text{ini}|t-1}^j \\ u_{|t-1}^j \end{bmatrix}, \tag{10}$$

where $u_{|t-1}^j$ denotes the optimal solution of the local DDPC for agent j at time $t-1$, wherein the summation is constant until the next time the DDPC is run. In addition, $(u_{\text{ini}|t-1}^j, y_{\text{ini}|t-1}^j)$ represents the past measurements of agent j at time $t-1$. This choice allows each local planner to run independently and results in the following network of distributed DDPCs (local QPs) to be solved at time t

$$\begin{aligned}\min_{(u^i,y^i)} \quad & \sum_{k=0}^{N-1} \left(\|y_k^i - y_k^{\text{des},i}\|_Q^2 + \|u_k^i - u_k^{\text{des},i}\|_R^2 \right) \\ \text{s.t.} \quad & y^i = \mathcal{G}_{i,i} \begin{bmatrix} u_{\text{ini}}^i \\ y_{\text{ini}}^i \\ u^i \end{bmatrix} + \sum_{j \neq i} \mathcal{G}_{i,j} \begin{bmatrix} u_{\text{ini}|t-1}^j \\ y_{\text{ini}|t-1}^j \\ u_{|t-1}^j \end{bmatrix} \\ & u_k^i \in \mathcal{U},\ y_k^i \in \mathcal{Y}, \quad k = 0, \dots, N-1, \end{aligned} \tag{11}$$

where $u_k^{\text{des},i}$ and $y_k^{\text{des},i}$ represent the desired inputs and outputs for agent i at the prediction step k (see Fig. 2). The optimal input and output trajectories are then passed to the low-level controller for tracking. In this work, we choose a subset of COM state variables for the output y while taking the ground reaction forces (GRFs) as the control inputs u. This will be clarified more in Section V-A. Consequently, the feasible set $\mathcal{U}$ is chosen as the linearized friction cone, i.e., $\mathcal{U} = \mathcal{FC} := \{\text{col}(f_x, f_y, f_z) | f_z > 0,\ |f_x| \leq \frac{\mu}{\sqrt{2}} f_z,\ |f_y| \leq \frac{\mu}{\sqrt{2}} f_z\}$, where μ denotes the friction coefficient.

IV. Nonlinear Low-Level Controller

The purpose of this section is to briefly provide the details of the low-level controller [40], based on QP and virtual constraints [41], to be used in this work. In particular, we model each robot using a floating base and represent the generalized coordinates of the system by $q \in \mathcal{Q} \subset \mathbb{R}^{n_q}$, where $\mathcal{Q}$ is the configuration space and n_q represents the number of degrees of freedom (DOFs) of the system. We further denote the joint-level torques by $\tau \in \mathcal{T} \subset \mathbb{R}^{m_\tau}$. In this notation, $\mathcal{T}$ denotes the allowable torques and m_τ denotes the number of actuators. It is assumed that the interaction wrenches between agents are encapsulated by the data-driven model. Therefore, we neglect these wrenches in the low-level controller. In particular, the overarching equations of motion for the full-order system of each agent become

$$M(q)\,\ddot{q} + H(q,\dot{q}) = \Upsilon\,\tau + J^\top(q)\,f, \tag{12}$$

where $M(q) \in \mathbb{R}^{n_q \times n_q}$ is 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 represents the GRFs at the stance feet. We further suppose that the positions of the stance leg ends, denoted by r, do not slip, i.e., $\ddot{r} = 0$.

With the dynamics in hand, we can now present the virtual constraints controller. In particular, we aim to track both force and COM trajectories generated by the high-level distributed planners. For this purpose, we consider virtual constraints as output functions to be regulated as $h(q,t) := h_0(q) - h^{\text{des}}(t)$. These virtual constraints are then imposed via partial feedback linearization [42]. Here, $h_0(q)$ denotes the controlled variables consisting of the COM position, orientation, and the Cartesian coordinates of the swing leg

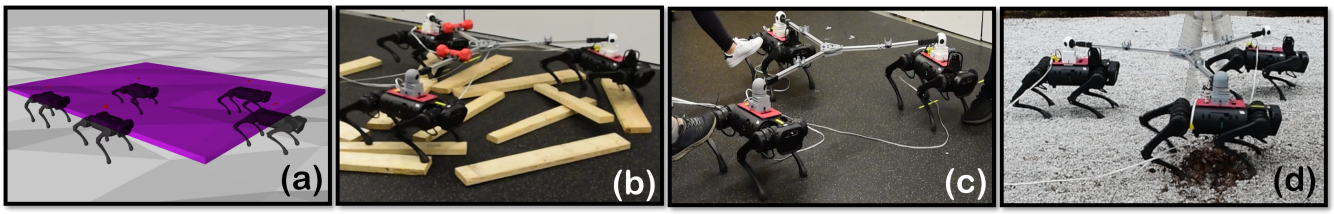

Fig. 3. (a) Simulation results of 5 agents over varying terrain with a payload of 10 (kg), (b) rough terrain experiment with unstructured wooden blocks and a 4.5 (kg) payload, (c) experiment with push disturbances, and (d) experiment maneuvering over gravel. Videos are available online [39].

ends. Finally, $h^{\text{des}}(t)$ represents the desired evolution of $h_0(q)$. In this work, the desired end position for a swing leg is chosen using the Raibert heuristic [43, Eq. (4), pp. 46], and the trajectory for the swing leg is defined using a Bézier polynomial. These virtual constraints, along with the no slippage condition, are concatenated into a single strictly convex QP to be solved at 1kHz as follows [40]

$$\begin{aligned} \min_{(\tau,f,\delta)} \quad & \frac{\gamma_1}{2}\|\tau\|^2 + \frac{\gamma_2}{2}\|f - f^{\text{des}}\|^2 + \frac{\gamma_3}{2}\|\delta\|^2 \\ \text{s.t.} \quad & \ddot{h}(\tau, f) = -K_P\, h - K_D\, \dot{h} + \delta \quad \text{(Output Dynamics)} \\ & \ddot{r}(\tau, f) = 0 \quad \text{(No slippage)} \\ & \tau \in \mathcal{T}, \quad f \in \mathcal{FC} \quad \text{(Feasibility)}, \end{aligned} \tag{13}$$

where γ_1, γ_2, and γ_3 are positive weighting factors, and the desired force profile $f^{\text{des}}(t)$ (i.e., inputs u) is prescribed by the high-level DDPC in (11). The equality constraints are expressed as 1) the output dynamics $\ddot{h} + K_D\,\dot{h} + K_P\, h = \delta$ for positive definite gain matrices K_P and K_D and δ being a defect variable to ensure feasibility, and 2) the no slippage condition $\ddot{r} = 0$. We remark that $\ddot{h}$ and $\ddot{r}$ are affine functions of (τ, f), hence, the problem is convex. We direct the reader to [17, Appendix A] for more information regarding the derivation of $\ddot{h}$ and $\ddot{r}$ according to Lie derivatives. The QP solves for the minimum-power torques τ while tracking the prescribed forces and COM trajectory.

V. Experiments

In this section, we provide the procedure for collecting the data for the reduced-order model, and further, provide the simulation and experimental results. Here we consider the 18-DOF quadruped A1 made by Unitree. The robot is modeled using a floating base, with the first 6 DOFs being composed of the unactuated position and orientation of the trunk. The remaining DOFs are composed of the actuated hip roll, hip pitch, and knee pitch joints for each leg. The robot weighs around 12.45 (kg) and the center of the trunk is 0.26 (m) above the ground during locomotion. We are, however, interested in multi-agent systems. We assume that each of the agents is holonomically constrained to other agents in a complete graph, i.e., there is no relative translational motion between agents. In order to accomplish this, we connect the robots together rigidly through a ball joint (see Fig. 3). In simulations, this is imposed via a distance constraint.

A. Data Collection

We begin by describing the I-O pairs considered in this work and the manner in which the data was collected. The data that is to be used in the Hankel matrix is collected in simulation at 100 (Hz) during nominal locomotion. This is in contrast to [36], which considered data collected on hardware. However, one of the contributions of this work is to create a reduced-order model of highly complex systems that interact with one another. From this standpoint, it makes sense to consider simulation data since each individual agent can be modeled accurately, but the complexity of collaborating agents is prohibitive in terms of defining a physics-based reduced-order model. This also shows good sim-to-real transfer for the learned model, as will be shown in what follows.

In order to perform the data collection in simulation, we first choose the inputs to be the forces at the contacting leg ends and we take the outputs as $y^{\text{d}} = \text{col}(\text{z}, \dot{\text{x}}, \dot{\text{y}}, \text{roll}, \text{pitch}, \omega_z)$, where z is the standing height, $\dot{\text{x}}$ and $\dot{\text{y}}$ are the linear velocities of the COM in the transverse plane, and ω_z denotes the angular velocity about the vertical axis of the torso. These outputs are chosen because they represent the variables of interest to an end-user when providing joystick commands to a robot. It should be noted that other I-O realization could provide fruitful results depending on the goal of the planner and the available measurements. During the data collection procedure, the quadrupeds are commanded to walk around in RaiSim [44] using just the low-level controller (13), while random noise is injected into the desired forces, which helps ensure the persistence of excitation. In particular, we choose the desired forces to be $u^{\text{des},i}_{k,\ell} := \text{col}(0, 0, \frac{m^{\text{net},i} g_0}{N^i_{c,k}})$ for each contacting leg $\ell \in \mathcal{C}^i_k$ and zero otherwise, where $m^{\text{net},i}$ is the total mass of agent i, g_0 is the gravitational acceleration, $N^i_{c,k}$ is the anticipated number of contacting legs for agent i at time k, and $\mathcal{C}^i_k$ is the anticipated set of contacting legs for agent i at time k. That is, when collecting data, we choose the desired force to be a random perturbation about the nominal amount of force that is required to hold the quadruped in a static position based on the number of anticipated contacts.

For this problem, we consider an estimation horizon of $T_{\text{ini}} = 10$ and $N = 25$ for the prediction horizon. We further ensure that T is chosen such that the amount of data collected far exceeds that required by the general theory. This, in turn, assists in potentially capturing more nonlinear information while also reducing the impact of noise. The collection of additional data when utilizing the formulation (5) could pose an issue due to an increase in decision variables. However, this is mitigated by utilizing the approximation (6).

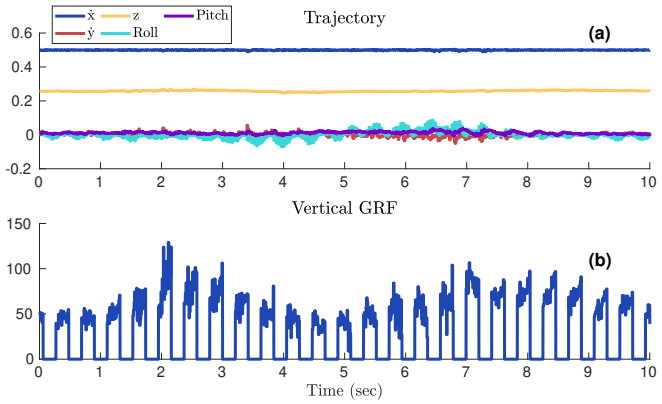

Fig. 4. The trajectory from the planner (a) and the vertical GRF of the front right leg from the planner (b) for agent 1. A forward speed of 0.5 (m/s) is commanded, and the standing height is 0.26 (m). The multi-agent system is subject to uneven terrain and a payload of 10 (kg), and can maneuver robustly. A snapshot of the simulation can be found in Fig. 3 (a).

B. Simulation Experiments

The high-level planner contains 450 decision variables per agent. In the centralized case using 5 agents shown here, that amounts to 2250 decision variables, which further motivates the necessity for a distributed approach. Under the distributed scheme, the predictive controller is updated every 40 (ms) (25 (Hz)), and the first 4 time steps are implemented, i.e., we predict over a horizon of 250 (ms) and implement the first 40 (ms) of the prediction. The high-level planner is solved using OSQP [45] and takes approximately 15 (ms) on an external laptop with an Intel® Core™ i7-1185G7 running at 3.00 GHz and 16 GB of RAM. However, solve times of $\sim$ 30 (ms) have been observed, further motivating the decision to update the planner at 40 (ms). The predictive controller parameters are chosen as $Q = \text{diag}(1e6, 1e5, 1e5, 2e5, 1e5, 1e4)$ and $R = I \otimes \text{diag}(0.05, 0.05, 0.5)$, where I is the identity of appropriate size, and $\otimes$ represents the Kronecker product. Finally, the parameters used by the low-level controller to track the trajectory and forces from the planner are chosen to be $\gamma_1 = 10^2$, $\gamma_2 = 10^3$, and $\gamma_3 = 10^6$, resulting in stable locomotion. All of the gains used in the simulation are the same as those used on hardware.

In order to show the efficacy of the proposed controller for 5 agents, we consider a compound experiment in simulation such that the multi-agent system is subject to an unknown payload of 10 (kg) and uneven terrain. A snapshot of the simulation can be found in Fig. 3 (a), while the prescribed forces and trajectory for the first agent can be found in Fig. 4. From these figures, it is evident that the planner produces forces that are feasible while also resulting in a viable COM trajectory for the low-level controller to track.

C. Hardware Experiments

Finally, we provide hardware experiments to show the effectiveness of the planner. In particular, for hardware experiments, we consider the use of 3 quadrupeds that are holonomically constrained using a ball joint. Snapshots of several experiments can be found in Fig. 3 (b)-(d), which shows the multi-agent system subject to external disturbances and unknown environments. In Fig. 5, we illustrate the behavior

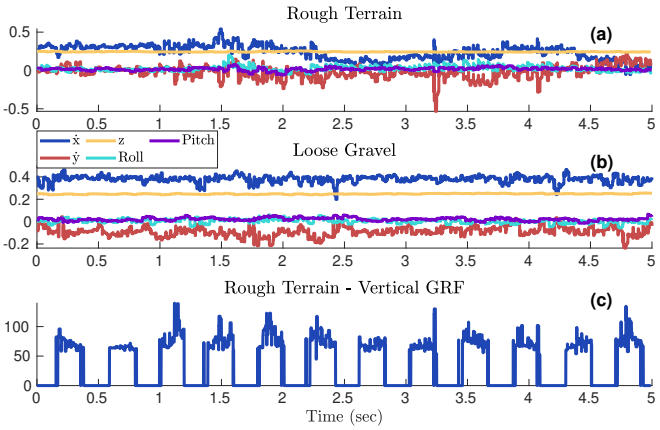

Fig. 5. The trajectory from the planner of agent 1 while trotting at approximately 0.4 (m/s) subject to (a) rough terrain with unstructured wooden blocks and (b) loose gravel. We further show the vertical GRF for the front right leg produced by the planner for rough terrain in (c). The GRF during the gravel experiment is similar.

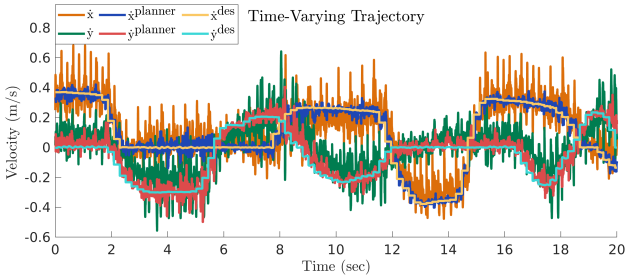

Fig. 6. The plot shows the tracking performance of the planner when using a time-varying trajectory provided using a joystick. In this experiment, the quadrupeds navigate flat ground subject to a 6.8 (kg) payload. The plot shows the trajectory of agent 1.

of the planner in terms of its prescribed trajectory when walking over unstructured wooden blocks (Fig. 5 (a)) and navigating over loose gravel (Fig. 5 (b)). The corresponding forces for the rough terrain experiment can be found in Fig. 5 (c). Finally, we provide an additional experiment to show the ability of the planner to track a time-varying trajectory subject to a 6.8 (kg) payload, where the trajectory produced by the planner can be found in Fig. 6. It is evident that the planner provides a robustly stable output even in the presence of significant uncertainty in terms of payloads and various environmental factors. Videos of the simulation and hardware experiments can be found in [39].

VI. Conclusion

This work presented a data-driven planner for robust multi-agent quadrupedal locomotion, wherein the robots were constrained to one another with a ball joint. We considered the use of behavioral systems theory to model the complex system and further proposed a distributed scheme to spread the computational load. We provided extensive experiments both in simulation and on hardware, which showed the robustness of this method to uncertainty in terrain, payloads, and external disturbances. Future work will examine this methodology when the robots do not form a complete graph, i.e., the agents are constrained but have limited freedom to change formation. Additionally, we will explore how this method could extend to an even greater number of agents.

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] P. Culbertson, J.-J. Slotine, and M. Schwager, "Decentralized adaptive control for collaborative manipulation of rigid bodies," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1906–1920, 2021.

[3] R. Murray, Z. Li, and S. S.S., *A Mathematical Introduction to Robotic Manipulation*. Taylor & Francis/CRC, 1994.

[4] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

[5] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems (RSS)*, 2013.

[6] D. Panagou, M. Turpin, and V. Kumar, "Decentralized goal assignment and trajectory generation in multi-robot networks: A multiple Lyapunov functions approach," in *IEEE International Conference on Robotics and Automation*, Hong Kong, China, June 2014, pp. 6757–6762.

[7] T. Machado, T. Malheiro, S. Monteiro, W. Erlhagen, and E. Bicho, "Multi-constrained joint transportation tasks by teams of autonomous mobile robots using a dynamical systems approach," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 3111–3117.

[8] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[9] F. Bullo, J. Cortés, and M. S, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton University Press, 2009.

[10] M. Mesbahi and E. M, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[11] A. Perizzato, M. Farina, and R. Scattolini, "Formation control and collision avoidance of unicycle robots with distributed predictive control," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 260–265, 2015.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[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," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 3, 2021.

[18] R. T. Fawcett, A. Pandala, J. Kim, and K. Akbari Hamed, "Real-Time Planning and Nonlinear Control for Quadrupedal Locomotion With Articulated Tails," *Journal of Dynamic Systems, Measurement, and Control*, vol. 143, no. 7, February 2021.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] Y. Ji, B. Zhang, and K. Sreenath, "Reinforcement learning for collaborative quadrupedal manipulation of a payload over challenging terrain," in *IEEE International Conference on Automation Science and Engineering*, 2021, pp. 899–904.

[27] 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.

[28] 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.

[29] I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor, *Exact and approximate modeling of linear systems: A behavioral approach*. Society for Industrial and Applied Mathematics, 2006.

[30] 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.

[31] 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.

[32] 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.

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

[34] 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.

[35] 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.

[36] R. T. Fawcett, K. Afsari, A. D. Ames, and K. Akbari Hamed, "Toward a data-driven template model for quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7636–7643, 2022.

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

[38] 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.

[39] Distributed data-driven predictive control for multi-agent collaborative legged locomotion. [Online]. Available: https://youtu.be/k1VsgvbR1Rs.

[40] 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*, vol. 6, pp. 1736–1741, 2021.

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

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

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

[44] 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.

[45] 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.