Hierarchical Control of Constrained Multi-Agent Legged Locomotion: A Data-Driven Approach

Randall T. Fawcett

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mechanical Engineering

Kaveh Akbari Hamed, Chair
Andrea L'Afflitto
Alexander Leonessa
Steve C. Southward

May 22, 2023
Blacksburg, Virginia

Keywords: Legged Locomotion, Real-Time Planning, Data-Driven, Nonlinear Control, Hierarchical Control

Hierarchical Control of Constrained Multi-Agent Legged Locomotion: A Data-Driven Approach

Randall T. Fawcett

(ABSTRACT)

The aim of this dissertation is to systematically construct a hierarchical framework that allows for robust multi-agent collaborative legged locomotion. More specifically, this work provides a detailed derivation of a torque controller that is theoretically justifiable in the context of Hybrid Zero Dynamics at the lowest level of control to produce highly robust locomotion, even when subject to uncertainty. The torque controller is based on virtual constraints and partial feedback linearization and is cast into the form of a strictly convex quadratic program. This partial feedback linearization is then relaxed through the use of a defect variable, where said defect variable is allowed only to change in a manner that is consistent with rapidly exponentially stable output dynamics through the use of a Control Lyapunov Function. The torque controller is validated in both simulation and on hardware to demonstrate the efficacy of the approach. In particular, the robot is subject to payload and push disturbances and is still able to remain stable. Furthermore, the continuity of the torque controller, in addition to robustness analysis of the periodic orbit, is also provided. At the next level of control, we consider emulating the Single Rigid Body model through the use of Behavioral Systems Theory, resulting in a data-driven model that adequately describes a quadruped at the reduced-order level. Still, due to the complexity and a considerable number of variables in the problem, the model further undergoes a 2-norm approximation, resulting in a model that is computationally efficient enough to be used in a real-time manner for trajectory planning. In order to test the method rigorously, we consider a series of experiments to examine how the planner works when using different gait parameters than that which was used during data collection. Furthermore, the planner is compared to the traditional Single Rigid Body model to test its efficacy for reference tracking. This data-driven model is then extended to the multi-agent case, where each agent is rigidly holonomically constrained to one another. In this case, the model is used in a distributed manner using a one-step communication delay such that the coupling between agents can be adequately considered while spreading the computational demand. The trajectory planner is evaluated through various hardware experiments with three agents, and simulations are also used to display the scalability of the approach by considering five robots. Finally, this dissertation examines how traditional reduced-order models can be used in tandem with data-based models to reap the benefits of both methods. More specifically, an interconnected Single Rigid Body model is considered, where the interaction forces are described via a data-driven model. Simulations are provided to display the efficacy of this approach at the reduced order level and show that the interaction forces can be reduced by considering them in the trajectory planner. As

in the previous cases, this is followed by experimental evaluation subject to external forces and different terrains.

Hierarchical Control of Constrained Multi-Agent Legged Locomotion: A Data-Driven Approach

Randall T. Fawcett

(GENERAL AUDIENCE ABSTRACT)

The goal of this dissertation is to create a layered control scheme for teams of quadrupeds that results in stable and robust locomotion, including a high-level trajectory planner and a low-level controller. More specifically, this work outlines an optimal torque-based whole-body controller that operates at the joint level to track desired trajectories. These trajectories are obtained by a high-level trajectory planner, which utilizes a data-driven predictive controller to create an optimal trajectory without explicitly requiring knowledge of a model. The hierarchical control scheme is then extended to consider collaborative locomotion. Namely, this work considers teams of quadrupeds that are rigidly connected to one another such that there is no relative motion between them. There are potentially large interaction forces that are applied between the robots that cannot be measured, which can result in instability. Furthermore, the models used to describe the interconnected system are prohibitively complex when being used for trajectory planning. For this reason, the data-driven model considered for a single robot is extended to create a centralized model that encapsulates not only the motion of a single robot but also its connection constraints. The resulting model is very large, making it difficult to use in a real-time manner. Therefore, this work outlines how to distribute the model such that each robot can locally plan for its own motion while also considering the coupling between them. Finally, this work provides one additional extension that combines a traditional physics-based model with a data-driven model to capitalize on the strengths of each. In particular, a physics-based model is considered as a baseline, while a data-driven model is used to describe the interaction forces between robots. In using this final extension, both improved solve times and smoother locomotion are achieved. Each of the aforementioned methods is tested thoroughly through both simulations and experiments.

Acknowledgments

First and foremost, to Dr. Kaveh Akbari Hamed. It has been a privilege to work with and learn from you over the course of the last 4 years. Needless to say, I would not be where I am now without the time, support, and countless opportunities that you provided to me during my studies. Although research has its ups and downs, you have not wavered in your support for both my success and well-being, and I am extremely grateful. You have continuously pushed me, and today I am all the better for it. Thank you.

I would also be remiss not to mention that working in the Hybrid Dynamic Systems and Robot Locomotion lab has been an amazing experience. Everyone brings a different set of skills to the table, and the collaboration with others in this lab is absolutely incredible. Working with everyone in this lab, including those who have since graduated, has been a fantastic experience. I could not ask for a better team to work with. Special thanks to all those who have contributed to this work, be it through debugging code with me, sifting through equations, or simply through stimulating conversation. All of the small moments have added up to make a profound impact on this work, and I am grateful.

Finally, I would like to thank both my friends and family. When times were hard, as they often can be as a graduate student, they supported me in more ways than I can express here. They were also there to celebrate with me when things went well. My friends and family have been a staple throughout my time as a graduate student, and I would not be finishing the final touches on this dissertation without them.

Thank you to everyone that helped along the way. It means the world.

Contents

List of Figures

List of Abbreviations

BST Behavioral Systems Theory

CBF Control Barrier Function

CLF Control Lyapunov Function

COM Center of Mass

COP Center of Pressure

DCM Divergent Component of Motion

DDPC Data-Driven Predictive Control

DeePC Data Enabled Predictive Control

DOF Degrees of Freedom

GRF Ground Reaction Forces

HZD Hybrid Zero Dynamics

IP Inverted Pendulum

ISS Input-to-State Stability

LIP Linear Inverted Pendulum

LTI Linear Time-Invariant

MPC Model Predictive Control

NLP Nonlinear Programming

NMPC Nonlinear Model Predictive Control

QP Quadratic Programming

RES-CLF Rapidly Exponentially Stabilizing Control Lyapunov Function

SLIP Spring Loaded Inverted Pendulum

SRB Single Rigid Body

VCs Virtual Constraints

ZMP Zero Moment Point

Chapter 1

Introduction

Human-centered societies are developed with human beings in mind, and in particular, for bipedal walkers capable of maneuvering over discontinuous terrains, including stairs and gaps in the environment. For this reason, traditional robots such as general wheeled vehicles can be inefficient when navigating many man-made structures. Conversely, legged robots maintain similar mobility, agility, and dexterity when compared to humans, allowing them to more efficiently navigate our environments. Although most structures are made with bipeds walkers in mind, it can also be observed that dogs and various other legged organisms possess the ability to maneuver in these environments as well. Quadrupeds, compared to their bipedal counterpart, take up more floor space but also possess superior stability properties and load-carrying ability, which motivates their use for accomplishing tasks alongside human beings. In particular, their enhanced stability yields an environment that is safer for humans, and their additional load-carrying capacity makes them more useful in terms of the number of tasks they can accomplish. Furthermore, a considerable number of legged robots have been created for this purpose (see Fig. 1.1), allowing researchers to develop and test algorithms quickly. However, simply having the ability to maneuver throughout various environments efficiently is only the first step in deploying legged robots to assist humans.

1.1 Motivation

As a society, human beings often work together in order to accomplish tasks. For example consider collaboratively transporting an object too difficult for one individual to move due to its size, geometry, or weight. While there are many applications in which a wheeled vehicle could assist in these types of tasks, these machines are not optimal when navigating human-centered societies and daunting terrain. In particular, the large wheeled robots are often only able to function in a factory or warehouse setting with little to no terrain obstructions. This motivates the development of collaborative-legged robots that can work together to assist humans and each other in complex and labor-intensive tasks—including cooperative transportation and manipulation—in challenging environments. While several quadrupeds have been equipped with robotic manipulators (see, e.g., Fig 1.1 (a) and (g)), there is a distinct gap in knowledge regarding the planning and control algorithms for quadrupedal robots collaborating with one another and with humans. In particular, collaborative loco-manipulation between multiple humans is simplified by the fact that we are able to perceive and under-

stand intent and adapt to the proprioceptively perceived external wrenches induced through collaboration. However, most applications of quadrupeds with manipulators broadly span two situations. The first is quadrupeds collaborating with a human in simplified scenarios wherein the person moves slowly and adapts to the motions of the robot to ease the interaction forces [7]. While this is a notable first step toward adequate collaboration with humans, forcing the human in the loop to react to and compensate for the robots movements is not ideal. Furthermore, unlike [7], most robotic arms that are lightweight and low cost do not have force measurements, which significantly inhibits robust control methods. This could also require very sophisticated estimation methods to determine the interaction wrench, which is particularly difficult in that most of the current state-of-the-art estimation techniques are not suitable for the hybrid nature of the floating base of quadrupeds subject to potentially large impulsive forces at impact. The latter situation is a single quadruped performing loco-manipulation, thereby eliminating inter-system wrenches. I.e., a single quadruped picking up an object for transport. In this case, the payload is often considered a disturbance and is not considered directly in the control law. While this has been shown to work well in practice for reasonably sized payloads, the control methodologies are not necessarily sufficient when moving to multi-agent loco-manipulation. Since loco-manipulation algorithms generally fall within these categories, this shows a fundamental gap in knowledge pertaining to robust control and planning methods that consider interactions for multi-agent legged robots, particularly in the cases where advanced sensors for force and torque measurements are unavailable.

In this work, we aim to examine how legged robots can effectively work together to accomplish a task. In this regard, consider a colony of ants. While transporting an object, they are able to collaborate, change configuration, and form complex structures allowing them to accomplish tasks that would otherwise be impossible. Here we use this as motivation for creating algorithms allowing multiple quadrupeds to work together in a constrained manner while considering the interactions between them using data-driven approaches. This work marks the first step toward enabling scalable loco-manipulation for robots, with an eventual goal of intelligent self-configurable collaboration between robots and with humans, much like how ants are able work together.

1.2 Scope, Goals, and Objectives

The ***overarching goal*** of this dissertation is to utilize and combine both physics-based and data-driven techniques in a hierarchical planning and control framework to control constrained multi-agent systems applied to teams of quadrupeds. In particular, the goals of this dissertation are enumerated as follows:

(1) Develop a computationally tractable nonlinear low-level controller that can is amenable to multi-agent control and task balancing and is theoretically justifiable in the context

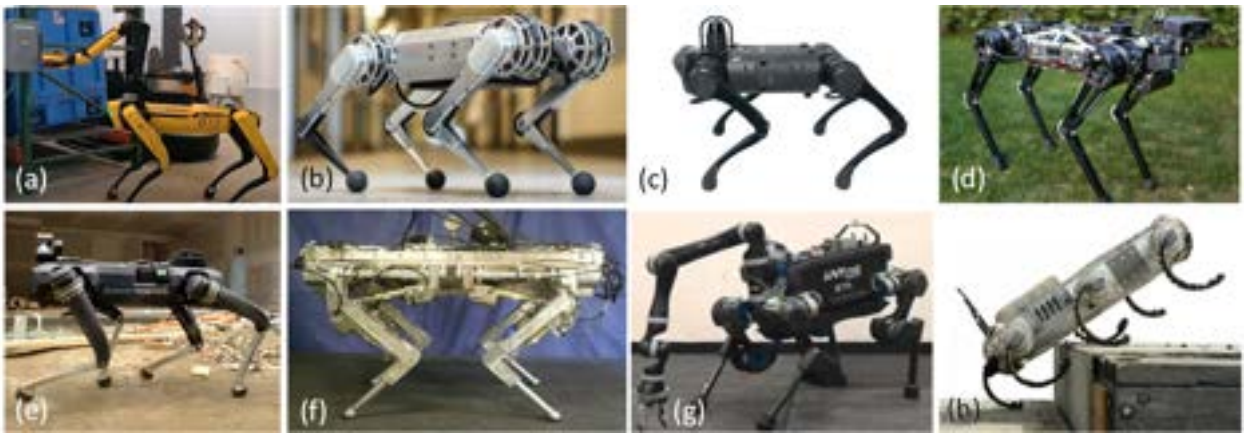

Figure 1.1: (a) Spot (Boston Dynamics) equipped with a robotic arm [1], (b) Cheetah Mini (MIT)[2], (c) A1 (Unitree) [3], (d) Cheetah 3 (MIT) [4], (e) Vision 60 (Ghost Robotics) [5], (f) HyQ (Italian Institute of Technology) [6], (g) ANYmal (ANYbotics)[7] equipped with a Kinova arm, (h) RHex (Boston Dynamics) [8].

of Hybrid Zero Dynamics.

(2) Development of reduced-order template models based on Behavioral Systems Theory for quadrupedal locomotion. This template model is then validated by incorporating it into a real-time predictive controller for online trajectory optimization. It is further integrated with the previously developed low-level controller to form a hierarchical control scheme.

(3) Create an extension of the data-driven predictive planner made for single agents to teams of holonomically constrained quadrupeds. This new planner is further distributed such that the computational burden can be reduced significantly, allowing this to be used in real time for trajectory planning.

(4) Provide rigorous simulation and experimental validation of the trajectory planner using data-driven template models on hardware for single and multi-agent systems. This validation shows the robustness of the proposed hierarchical framework when subject to disturbances and various unknown terrains, including unstructured indoor environments and different outdoor surfaces.

(5) Combine both data-driven and physics-based reduced order models to capture the most important properties of both models. While the data-driven models have proven to be successful when applied to multi-agent collaborative legged locomotion, we aim to improve performance by augmenting traditional model-based approaches with Behavioural Systems Theory to improve the model while maintaining computational efficiency by again considering a distributed approach.

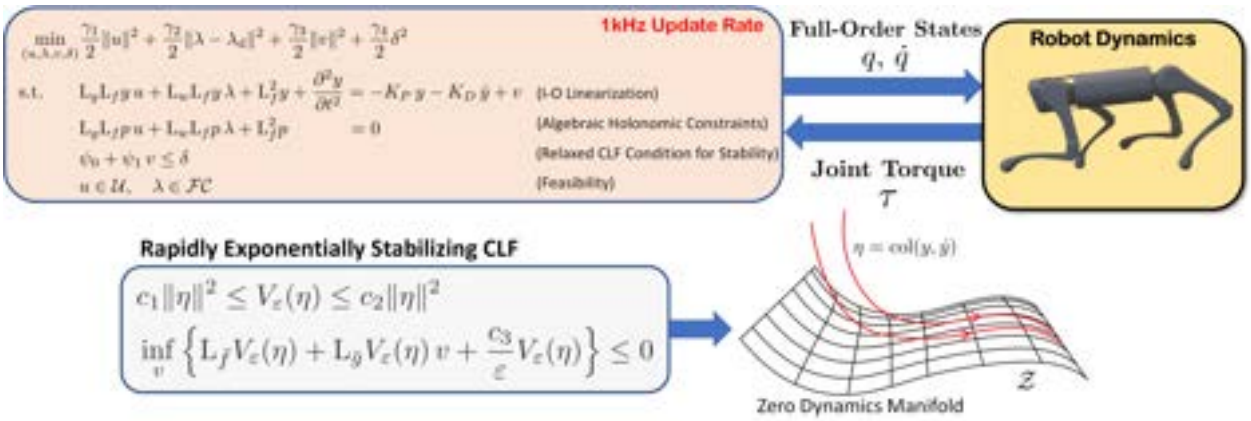

Figure 1.2: Overview of **R1**.

(6) Provide additional simulation and experimental validation of the hybrid physics- and data-based algorithm developed for a team of quadrupedal robots that are holonomically constrained. Similar to the previous case, various disturbances and terrains are considered to evaluate the robustness of the approach.

These objectives are organized into the following specific research objectives:

(R1) Research Aim 1: Nonlinear Low-Level Control based on Quadratic Programming, Virtual Constraints, and Control Lyapunov Functions: This research aim develops, theoretically justifies, and experimentally implements an optimization-based nonlinear control methodology for stabilizing quadrupedal locomotion. This framework utilizes virtual constraints and Control Lyapunov Functions (CLFs) in the context of Quadratic Programs (QPs) to robustly stabilize periodic orbits for hybrid models of quadrupedal robots (see Fig. 1.2). Properties of the proposed QP are studied wherein sufficient conditions for the continuous differentiability of the controller are presented. Additionally, this dissertation addresses the robust stabilization problem of the orbits based on the Poincaré sections analysis and input-to-state stability (ISS). The proposed controller is numerically and experimentally validated on the A1 quadrupedal robot with 18 Degrees of Freedom (DOF) to demonstrate the robust stability of trotting gaits against external disturbances and unknown payloads.

(R2) Research Aim 2: Data-Driven Template Model and Trajectory Planner for a Single Quadruped: This research aim 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 traceability when compared to the full-order nonlinear model. The spirit of the trajectory planning approach in this work draws on recent advancements in the area of Behavioral Systems Theory (BST). 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 without

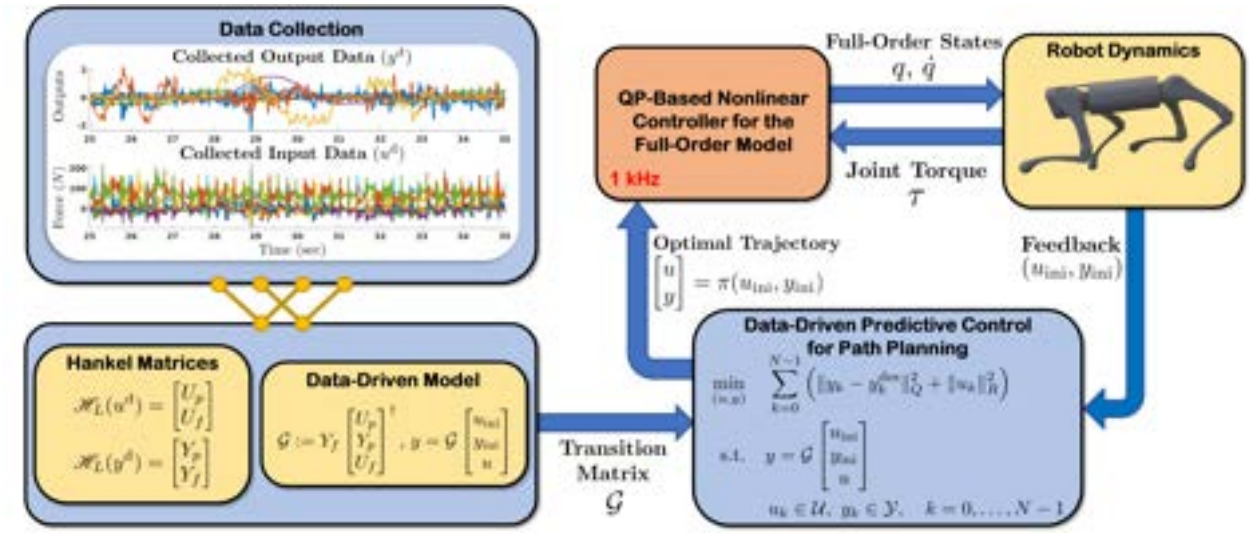

Figure 1.3: Overview of **R2**.

sacrificing tractability. In particular, this work considers input and output states similar to that of the Single Rigid Body (SRB) model and proceeds to develop a data-driven representation of the system, which is then used in a predictive control framework to create an optimal trajectory for quadruped to follow. The optimal trajectory is passed to the low-level and nonlinear model-based controller developed in **(R1)** to be tracked while considering the full-order model. An overview can be found in Fig. 1.3. Experimental results are provided to establish the efficacy of this hierarchical control approach for trotting and walking gaits of a single high-dimensional quadrupedal robot on unknown terrains and in the presence of disturbances. Furthermore, a comparison is provided to show the effectiveness of this approach compared to a model-based approach, as well as an experiment showing the use of this planner in tandem with the manufacturer's built-in low-level controller to emphasize the agnostic nature of the approach.

(R3) Research Aim 3: Extension of Data-Driven Template Models and Trajectory Planners for Constrained Multi-Agent Systems: This research aim extends the results of task **(R2)** by creating 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 for trajectory planning based on BST to model the sophisticated and high-dimensional structure that includes the holonomic constraints induced by the interconnection of subsystems. The resulting model is then used in tandem with distributed control techniques such that the computational burden is shared across agents while the strong coupling induced by interaction forces 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 (see Fig. 1.4). This research aim is evaluated in simulation with up to five agents to examine the scalability of the approach and is further experimentally validated on three A1 quadrupedal robots subject to various uncertainties,

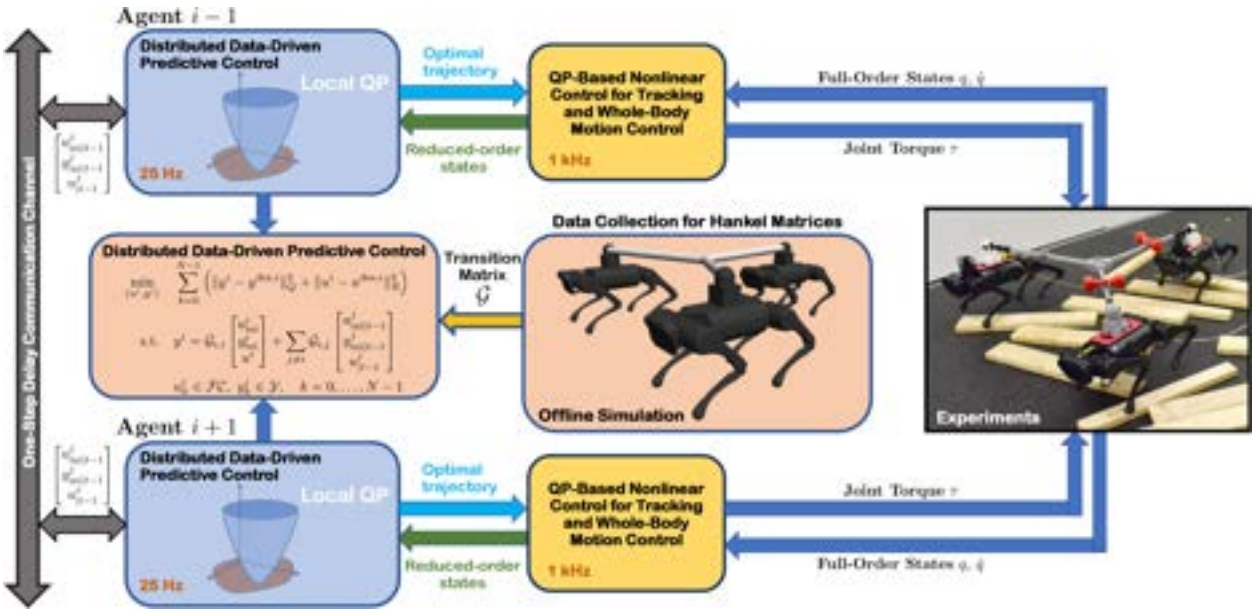

Figure 1.4: Overview of **R3**.

including payloads, rough terrain, and push disturbances.

(R4) Research Aim 4: Combining Physics-Based and Data-Driven Reduced-Order Models for Constrained Multi-Agent Systems: the primary objective of this research aim is to create a computationally tractable planner for constrained locomotion of teams of robots when combining physics-based and data-driven methods during the trajectory planning phase. In particular, the data-driven planners are shown to be effective but lack several important properties, including sparsity of the optimization problem, which can cause difficulties for practically robust and performant implementations of the algorithm. For this reason, the aim of this section is to maintain practical performance by considering traditional model-based techniques for each individual agent while modeling the interaction wrenches using a data-driven approach, as shown in Fig. 1.5. In doing so, we are able to maintain a high degree of sparsity, considerably decreasing the computation time for each agent, particularly in the case that the algorithm is distributed similarly to the previous research aim. This approach is finally validated at both the reduced- and full-order level in simulation and is further evaluated on hardware to establish the efficacy of the approach.

1.3 Literature Review

The purpose of this section is to outline pertinent research in the areas of modeling, trajectory-planning, and control of legged robots, various data-driven approaches and their applications to robotics, and control of multi-agent systems.

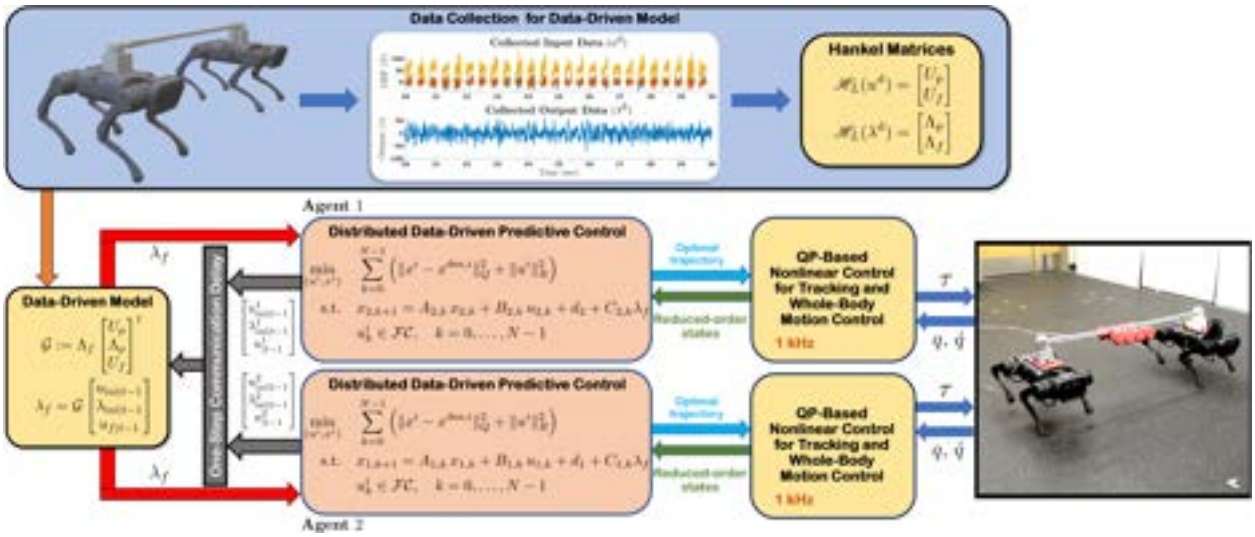

Figure 1.5: Overview of **R4**.

1.3.1 Full-Order Model of Legged Robots

The full-order modeling of a legged robot is slightly more complex than a general robotic system. In particular, legged robots are *hybrid* in that they exhibit both continuous- and discrete-time dynamics [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. The system undergoes the continuous-time dynamics throughout a particular domain, where a domain is defined by a particular set of contact points with the environment. Upon a new foot making or breaking contact, the system undergoes the discrete-time dynamics [33]. More specifically, when a leg makes contact with the ground, the velocity of the system is subject to a rapid change in velocity, which is encapsulated by the discrete-time dynamics. However, the continuous-time domains, similar to most other robotic systems, can adequately be described using the Euler-Lagrange method. A variety of approaches have since considered the hybrid model, leading to incredibly robust locomotion even when subject to uncertainty.

1.3.2 Nonlinear Controllers

In this section, we outline some of the many exceptional control methodologies that consider hybrid models of legged locomotion. In particular, some of the state-of-the-art approaches include controlled symmetries [34], hybrid reduction [35, 36], transverse linearization [37, 38], and Hybrid Zero Dynamics (HZD) [21, 39, 40, 41, 42, 43, 44, 45]. Alternatively, some other methods consider more traditional control techniques previously used in robotics for whole-body task space control [46, 47], although these methods do not explicitly consider the hybrid nature of locomotion in the development of the control law. Of these methods that

consider the hybrid nature of locomotion, only HZD and transverse linearization are able to directly consider the general case of underactuation. This does not necessarily pose an issue depending on the robot and the gait considered. For example, some bipeds with non-trivial feet are fully actuated, and quadrupeds are fully actuated when at least two legs are on the ground. Although, this does not necessarily mean that the contact Jacobian will be full rank, which could still prove to be problematic for some approaches. However, having the ability to operate in underactuated configurations, even for fully actuated systems, can introduce more natural and dynamic motions.

In this work, we primarily focus on the HZD methods for the development of nonlinear low-level controllers. Under this methodology, the limbs are coordinated through the use of holonomic kinematic constraints, generally referred to as Virtual Constraints (VCs) [12, 40]. These virtual constraints can be effectively tracked through the use of Input-Output (I-O) feedback linearization [48]. Furthermore, VCs have been used in a variety of contexts for legged locomotion, including both bipeds [23, 29, 40, 41, 44, 49, 50, 51, 52, 53, 54] and quadrupeds [43, 55, 56, 57, 58, 59, 60, 61]. In addition to legged robots, VCs have been successfully implemented for both prostheses and exoskeletons [62, 63, 64, 65, 66, 67, 68], which demonstrates the versatility of the approach to a variety of applications.

While the HZD method has been very successful, the method traditionally requires offline computations for trajectory generation. In the early applications, the trajectory of the VCs were created using a Nonlinear Programming (NLP) problem [29, 69, 70, 71, 72, 73, 74] using existing solvers. These methods result in VCs that produce a stable periodic orbit for locomotion. Most of these methodologies require simulating a set of VCs forward in time to determine if they produce a stable orbit, making them unrealistic for real-time planning due to high computational demand [29, 75]. Using offline optimization for VCs has resulted in stable locomotion [29, 58, 75, 76, 77] that has proven to be stable even in the presense of some degree of uncertainty, but does not consider environmental factors or gait initiation and termination. Therefore, it is less desirable for robust locomotion subject to considerable uncertainty and disturbances. This motivates the development of hierarchical control methodologies and computationally tractable planners such that HZD methods can be used adequately in real time.

Finally, HZD methods have the possibility of becoming numerically unstable in some scenarios. In particular, this method often relies on I-O linearization, which requires inverting a decoupling matrix. In the case of legged locomotion, this matrix is often ill-conditioned—and, in some situations, singular or otherwise not invertible. For this reason, the HZD methods have often been combined with computationally tractable QP in order to overcome these issues [41, 58, 60, 61, 78, 79, 80, 81]. The I-O linearization is used as a constraint in the QP with the addition of a defect variable, thereby bypassing many of the potential numerical issues as will be discussed in greater detail in Chapter 2. This further allows one to create gaits that are overactuated, ensure that Ground Reaction Forces (GRFs) and torques are feasible, and impose Control Lyapunov Function (CLF) conditions on the control law for further theoretical guarantees [41, 78, 79, 80, 81, 82]. The inclusion of a CLF is particularly

interesting in that it can provide a dynamic balance between conflicting goals of different tasks and has the potential to create more natural motions during loco-manipulation [78]. When formulated in a centralized manner, the CLF conditions can also provide theoretical guarantees regarding the interaction between different subsystems [82]. While this provides an interesting potential avenue for collaborative legged locomotion, centralized control laws tend not to be scalable to large groups of robots, making the method intractable for large-scale multi-agent systems. We further suppose that it is more valuable to consider the multi-agent interactions at the planning level to avoid abstraction and computational issues at the low-level of the control scheme.

1.3.3 Model Predictive Control

As an alternative to nonlinear controllers, a significant amount of research has been conducted regarding the use of linear Model Predictive Control (MPC) in the context of legged robots [57, 61, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92]. MPC controllers usually run at a slower rate ($50-500$ Hz) and act primarily as a trajectory planner. The resulting trajectory is then tracked using a lower-level controller, ranging from simple Jacobian mappings to sophisticated control methodologies such as those presented above. Unlike the nonlinear control methodologies presented in the previous section, most of the MPC approaches do not use full-order kinematics or dynamics and instead require a reduced-order model of some kind. Some notable exceptions include [91, 92], but in this work, we solely focus on methods that utilize reduced-order models.

Inverted Pendulum

The Linear Inverted Pendulum (LIP) model is among the older reduced-order models but is still used and researched primarily due to its successful applications and simplicity [86] (see Fig. 1.6). The traditional LIP model considers the body of a robot as a point mass located at a constant height above the ground and states that the Center of Mass (COM) evolves in the transverse plane according to inverted pendulum dynamics. In order to use this model successfully for locomotion, one must consider the Zero Moment Point (ZMP). In particular, stable locomotion dictates that the COM and Center of Pressure (COP) trajectories must evolve in a manner such that the COP remains within the convex hull formed by the contacting feet at all times. While not particular to the LIP model, there is an additional constraint that the GRF must remain feasible at all times, where this can be described by the net GRF in the case of multiple

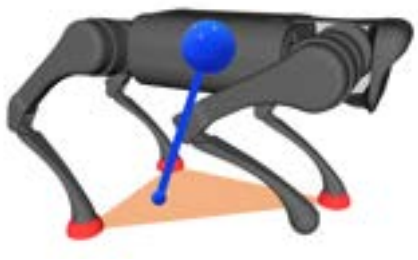

Figure 1.6: Representation of the standard LIP model.

contacting legs. When combined with MPC techniques, the LIP model is able to produce stable trajectories for robots in a computationally tractable manner. This simple but effective model has been implemented on multiple robots showing the versatility of the approach [57, 83, 84, 85, 86, 93, 94]. One of the benefits of this model is that it can be directly added to a MPC without additional linearization or model abstraction, aside from the constant height assumption. However, due to the ZMP requirement, the standard LIP model produces trajectories that are generally quasi-static. While the results are stable, dynamic legged locomotion is effectively encapsulated by a series of successive controlled "falls" that result in a stable periodic orbit for the system. Therefore, in order to obtain dynamic and agile locomotion, the COP must leave the support polygon for some period of time. The LIP model is further restrictive since it assumes a constant height and constant lumped mass and inertia.

A close relative to the LIP model, namely the Spring Loaded Inverted Pendulum (SLIP) model [95], has also been used in various applications. Although the SLIP model retains many of the desirable properties of the LIP model, it is nonlinear. The nonlinearity of the model comes from the fact that the COM can oscillate subject to a spring, which better encapsulates the nature of locomotion, particularly for bipeds. This is beneficial in that it often leads to more dynamic and energy-efficient gaits, but the downfall is that it requires successive linearization or the use of Nonlinear MPC (NMPC). In turn, this leads either to further model abstraction or an increased computational burden. Still, researchers have obtained very good results with this model [22, 95, 96, 97, 98]. One additional concept that extends the LIP model is that of the Divergent Component of Motion (DCM) [99, 100, 101, 102]. While the LIP model only captures the COM evolving in the transverse plane, the DCM is effectively the LIP model in 3 dimensions. Unlike the SLIP model, the DCM still remains linear and can be integrated directly with standard MPC formulations subject to additional decision variables and constraints when compared to the LIP.

Even though all of the above models have proven to be successful for general locomotion, they all share one considerable disadvantage in the context of collaborative legged locomotion: they do not consider moments about the COM. More specifically, external forces acting on the COM can be considered adequately, but applied torques must be neglected [103, 104]. Furthermore, the forces applied externally do not always have a line of action passing through the center of mass. Therefore, they induce a moment on the body that cannot be considered during trajectory planning. This poses an issue and implies that the Inverted Pendulum (IP) models will be insufficient in the context of this work as we aim to create planners that can correctly consider the entire interaction wrench.

Single Rigid Body

More recently, the legged locomotion community has begun to adopt the Single Rigid Body (SRB) model [4] (see Fig. 1.7). Using this model, a robot is approximated only by its trunk

and is propagated through space by forces acting on the system some distance away from the COM. This model has been used in a large portion of the literature on legged robots and has produced many successful implementations both in simulation and on hardware [4, 89, 90, 105, 106, 107, 108].

There are numerous advantages to using this model, particularly in that it is amenable to dynamic locomotion, and it can consider torques induced about the center of mass innately. In addition, contrary to the IP models, the inputs to the SRB model are the GRFs. Using the GRFs as inputs is extremely beneficial since it is then simple to constrain the GRF on each leg such that it always falls within the friction cone, as opposed to the IP methods wherein the best one can achieve is to consider the net force. However, one difficulty is that it is a nonlinear model and also has a rotation matrix as a state of the system. When used in tandem with NMPC, this does not necessarily pose an issue, but it is often desirable to cast planning and control problems into the form of a convex QP for computational speed. Furthermore, a *strictly* convex QP is also guaranteed to have a single solution, which is also desirable. For this reason, it is common to linearize the model in a successive manner using the small angle approximation [4] or variational-based linearization [89, 90]. Both methods have proven to be extremely robust on various hardware platforms, primarily in the context of quadrupedal locomotion. It is worth mentioning that the variational approaches have been used more when considering very dynamic motions. This is because variational-based linearization does not suffer from singularities when passing through angles that would traditionally result in gimbal lock. This could also be avoided by using a quaternion-based approach as has been done on drones [109], but using an Euler representation is still far more common [110]. One additional concern is that the SRB model does not consider the mass of the legs of the robot. When using small robots such as A1 and Mini Cheetah (see Fig. 1.1), the mass of the legs is very small compared to that of the trunk. Therefore, it is reasonable to assume the legs are massless for these small-scale quadrupeds. However, for larger robots performing dynamic motions, such as Vision60 and Anymal, the mass of the legs play a more considerable role in the overall dynamics of the system. For this reason, some researchers have adopted the use of gravity compensation for the legs to somewhat mitigate the problem [89, 111], although it would be desirable to consider them in some manner while planning if the mass of the legs proves to be considerably impactful on the overall system performance.

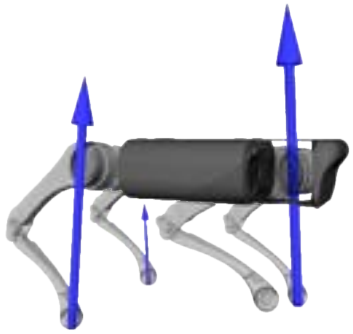

Figure 1.7: Representation of the SRB model. The blue arrows represent the GRF, and the translucent bodies (the legs) are neglected in the model.

1.3.4 Multi-Agent Planning and Control

A considerable amount of research has been conducted on multi-agent systems, including UGVs [112, 113, 114], UAVs [115, 116, 117, 118, 119], and various other applications like cooperative manipulation [120, 121]. When considering the formation control problem of multi-agent systems, it is often necessary to use a reduced-order model, similar to that which is done by the legged locomotion community when utilizing MPC. However, it is common in the context of formation control to consider the use of a kinematic or kino-dynamic model [117, 122, 123, 124, 125], where both kinematics and dynamics are considered to some degree.

Using a reduced-order model is a considerable benefit in terms of computational burden, particularly in the case of centralized control and planning for many agents. There are several very common models used for navigation, including the single integrator, double integrator, and unicycle models, in order of increasing complexity [117, 123, 124, 126]. These methods work well when the systems are not constrained to one another, particularly in the case of inherently stable systems such as wheeled vehicles. When moving to more elaborate unstable systems such as drones and quadrupeds, these models become less illustrative of the actual motions exhibited by the system. Fortunately, in the case that the trajectory can be tracked sufficiently well by a low-level controller, theoretical guarantees can still be made about the system as a whole [126]. In this sense, using a reduced-order kinematic model may not represent the system perfectly, but the analysis is simplified and formal guarantees regarding the stability of the system are still possible. Finally, kinematic models are not amenable to considering external wrenches applied to the system. This is a particular issue when dealing with holonomically constrained multi-agent systems (see, e.g., Fig. 1.8) with considerable interaction forces occurring between agents. In fact, very little research has been conducted on trajectory planning for robotic systems subject to holonomic constraints during collaboration and formation control. This is particularly the case for systems that are inherently unstable mobile robots, such as drones and quadrupeds, and those that exhibit strong interaction forces. However, a few recent and notable exceptions include [103, 127, 128], both of which are model-based. In particular, [127] considers a linearization approach to consider the interaction forces, while [128] utilizes a NMPC formulation as opposed to linearizing.

Figure 1.8: Example of a constrained group of quadrupeds.

It is also worth discussing different control methodologies for multi-agent systems. In particular, multi-agent control has been accomplished using Control Barrier Functions (CBFs) [129, 130, 131, 132, 133, 134], artificial potential fields [135, 136], game theory [137, 138, 139, 140, 141, 142], and reinforcement learning [143, 144, 145] to name a few. Of these methods, the use of CBFs is becoming increasingly common. This is partly due to the fact that they

are typically simple to implement and theoretically justify. In particular, the CBF enforces that each agent stays within user-defined admissible sets. In addition, CBFs can be used to accomplish other tasks as well, such as collision avoidance between agents and environmental obstacles. CBFs also have a large number of similarities to potential fields, making them an attractive choice for planning [146]. Another conceptually similar methodology to CBFs is funnel control [147]. Using funnel control, one is able to restrict the output of a system to a time-varying set, where that set can converge over time to the desired output, providing considerably more flexibility when compared to CBFs. Furthermore, restrictions can be placed on the transient dynamics as well. While this method has not gained considerable traction in robotics, the theory is well-developed and has been used in several applications for multi-agent systems, particularly in recent years [148, 149, 150]. On the other hand, game theory takes a very different approach compared to CBFs and funnel control. From a game theoretic perspective, it is beneficial to treat each agent (player) as either a teammate with which to cooperate or an adversary with which to compete. In formation control, the former is more important. Each player is responsible for accomplishing a task—for example, tracking a forward velocity—while the group is responsible for maintaining formation. Consider the scenario where an agent is falling behind in the formation or is perturbed considerably. When using a CBF, it becomes the sole responsibility of the perturbed agent to come back into formation. In contrast, a game theoretic algorithm would allow other agents to move in a manner to *help* the perturbed agent to get back into formation, while also considering their own goals. In this sense, game theory provides a more holistic view of the group to accomplish a task. However, real-time implementation of game theoretic algorithms has been a barrier to the wide-scale adoption of the methods.

Lastly, when considering many agents, the system is high-dimensional, which also motivates distributed approaches when possible [60, 118, 127, 136, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162]. These methods reduce the computational burden for controllers and planners, but they have also *not* been readily extended to legged robotic systems with high-dimensional hybrid nonlinear dynamics and unilateral constraints. Furthermore, many methods used for distributed systems assume weak interactions between the agents [155, 156, 157, 158, 159, 160, 161], which is not the case in this work since the robots are physically constrained to one another as opposed to enforcing fictitious virtual constraints to maintain a formulation. There has been limited work that directly considers the nonlinear interactions between subsystems in a distributed approach [60, 127, 128], which demonstrates the gap in knowledge pertaining to systems with large coupling terms.

1.3.5 Data-Driven Methods

Data-driven methods are becoming more popular as systems become increasingly complex [163]. This motivates their use in multi-agent systems, especially in the case where the individual agents are dynamically complex, such as quadrupeds. Even though individual systems can be modeled adequately, their combined dynamics make the system consider-

ably more sophisticated, and an effective and efficient reduced-order model has yet to be determined. One such method recently popularized in the robotics community is reinforcement learning [61, 164, 165, 166]. This method of learning has been used for collaborative locomotion among multiple quadrupeds [167] 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 that is not based on learning and instead considers a direct system identification method.

Here, we consider the use of data-driven methods in the context of Behavioral Systems Theory (BST). This methodology parameterizes a Linear Time-Invariant (LTI) system directly in terms of its measured trajectories [168, 169, 170]. BST, 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 [154, 171, 172, 173, 174, 175, 176, 177]. Recently, extensions have been made to use such methods for certain types of nonlinear systems [178] and linear parameter varying systems [179], 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 [154, 172, 173, 174, 177]. We further note that there have been extensions to hybrid classes of systems as well [180, 181], but none of these methods explicitly apply to *nonlinear* hybrid systems, nor are they concerned with periodic orbits, as is the case when dealing with quadrupeds. To the best of the authors' knowledge, there have not been implementations for multi-agent systems using BST prior to this work. We further note that the extension to collaborative legged locomotion introduces many difficulties, including the hybrid nonlinear nature of legged locomotion and the inclusion of unilateral constraints, while also requiring that the GRF is feasible for each leg relative to both the friction cone and the max force that the full-order system is able to apply.

1.4 Relevant Publications

As part of this research, two journals and a conference paper have been written, including publications in IEEE Robotics and Automation Letters (RAL), IEEE Control Systems Letters (LCSS), and IEEE International Conference on Robotics and Automation (ICRA). In addition, **[J1]** was presented at the IEEE American Controls Conference, and **[J2]** was presented at the IEEE International Conference on Intelligent Robots and Systems (IROS). It is also worth mentioning that **[C1]** has been nominated for the Outstanding Paper Award in Multi-Robot Systems for ICRA 2023. Chapters 2, 3, and 4 are primarily composed of the material found in **[J1]**, **[J2]**, and **[C1]**.

[**J1**] **R. T. Fawcett**, A. Pandala, A. D. Ames, and K. Akbari Hamed, "Robust stabilization of periodic gaits for quadrupedal locomotion via QP-based virtual constraint controllers," IEEE Control Systems Letters, pp. 1736-1741, 2021

[**J2**] **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.

[**C1**] **R. T. Fawcett**, L. Amanzadeh, J. Kim, A. D. Ames, and K. Akbari Hamed, "Distributed Data-Driven Predictive Control for Multi-Agent Collaborative Legged Locomotion," IEEE International Conference on Robotics and Automation, 2023.
Finalist for the Outstanding Paper Award in Multi-Robot Systems

In addition to the publications listed above, I have also contributed to several other journal and conference publications throughout my time as a graduate student in IEEE Transactions on Robotics (TRO, in review), ASME Journal of Dynamic Systems, Measurement, and Control (DSMC), IEEE Robotics and Automation Letters (RAL), IEEE Control Systems Letters (LCSS), and IEEE International Conference on Intelligent Robots and Systems (IROS). Furthermore, in what follows, [**J4**] and [**J5**] were presented at the IEEE International Conference on Intelligent Robots and Systems (IROS) and the IEEE Conference on Decision and Control (CDC), respectively, in addition to the primary publication. Finally, [**J6**] received the 2022 ASME DSCD Rudolf Kalman Best Paper Award. The additional publications are provided as follows:

[**J3**] J. Kim, **R. T. Fawcett**, V. R. Kamidi, A. D. Ames, and K. Akbari Hamed, "Layered Control for Cooperative Locomotion of Two Quadrupedal Robots: Centralized and Distributed Approaches," IEEE Transactions on Robotics, *In Review, March 2023*.

[**J4**] A. Pandala, **R. T. Fawcett**, U. Rosolia, A. D. Ames, and K. Akbari Hamed, "Robust Predictive Control for Quadrupedal Locomotion: Learning the Gap between Reduced- and Full-Order Models," *IEEE Robotics and Automation Letters*, Vol. 7, Issue 3, pp. 6622-6629, July 2022.

[**J5**] V. R. Kamidi, J. Kim, **R. T. Fawcett**, A. D. Ames, K. Akbari Hamed, "Distributed Quadratic Programming-Based Nonlinear Controllers for Periodic Gaits on Legged Robots," *IEEE Control Systems Letters*, vol. 6, pp. 2509-2514, 2022

[**J6**] **R. T. Fawcett**, A. Pandala, J. Kim, and K. Akbari Hamed, "Real-time planning and nonlinear control for quadrupedal locomotion with articulated tails," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 143, Issue. 7, pp. 071004-1-071004-15, Jul, 2021.
This work received the 2022 ASME DSCD Rudolf Kalman Best Paper Award

[C2] J. Martin, V. R. Kamidi, A. Pandala, **R. T. Fawcett**, and K. Akbari Hamed, "Exponentially stabilizing and time-varying virtual constraint controllers for dynamic quadrupedal bounding," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3914-3921, Las Vegas, NV, October 2020

1.5 Dissertation Outline

This dissertation is broken down into three primary parts. In particular, Part I consists of research task **(R1)** and details the low-level torque controller that is used in the overarching hierarchical framework that is laid out in Part II. More specifically, Part II focuses on research tasks **(R2)** and **(R3)**, which outlines a tractable data-driven path planner for single and multi-agent quadrupedal systems. The last technical portion of this dissertation, Part III, discusses research task **(R4)** and provides the experimental validation of the hybrid physics- and data-based method for constrained collaborative legged locomotion. Finally, the dissertation is brought to a close in Chapter 6 with a summary of the work followed by directions for future research.

Part I

Nonlinear Whole-Body Motion Control

Chapter 2

QP-Based Virtual Constraint Controllers with Lyapunov Functions

2.1 Introduction and Motivation

One major drawback of QP-based nonlinear control approaches is the possible lack of continuous differentiability (i.e., $\mathcal{C}^1$) of the feedback laws with respect to the system's state [182]. While a lack of $\mathcal{C}^1$ continuity does not imply that the control law will not provide adequate performance, the analysis can potentially become more difficult. Practically, the stability analysis of periodic locomotion can be checked via the eigenvalues of the Poincaré map [40, 183] which requires $\mathcal{C}^1$ continuity of the feedback laws with respect to the state. Hence, the lack of smoothness of the feedback laws prohibits the use of the powerful Poincaré sections analysis tools to study the stability of gaits. This motivates the creation of a high-performance control law that is $\mathcal{C}^1$.

The *overarching goal* of this chapter is to present a continuously differentiable and QP-based nonlinear controller, based on virtual constraints and Control Lyapunov Functions (CLFs), to robustly stabilize hybrid periodic orbits for quadrupedal locomotion. The objectives and contributions of this work are as follows. We study the properties of the proposed QP-based nonlinear controller and present sufficient conditions under which the feedback laws become $\mathcal{C}^1$. We investigate conditions under which the orbit is invariant for the closed-loop hybrid system. The robust stability properties of the periodic orbit under the proposed nonlinear controller are studied via the Poincaré sections analysis and input-to-state stability (ISS). We numerically and experimentally validate the proposed nonlinear controller on the advanced A1 quadrupedal robot with 18 DOFs to demonstrate the stability and robustness of trotting gaits against unknown payloads and external disturbances. To the best of the authors' knowledge, this is the first time a full-order HZD controller with exact feedback linearization has been implemented on quadrupedal robots. The use of a model-based CLF for locomotion has only been validated on bipedal platforms [41], making this the first application of a model-based CLF to quadrupedal locomotion, particularly when used in the HZD framework. In this work, the controller developed in this chapter will generally be referred to as the low-level controller. The majority of the contents of this chapter are taken from our previous work [43].

Previous work in the HDSRL lab [57] has used QP-based nonlinear controllers for numerical

simulations of quadrupedal locomotion without studying the $\mathcal{C}^1$ continuity, invariance of the orbit, and robust stability. Furthermore, previous work [184] had conducted a hardware implementation of a QP-based virtual constraint controller subject to a Lyapunov function, but this work lacked the theoretical foundation presented here. The current work technically addresses these properties while experimentally evaluating the control framework on hardware. This work also differs from [182] and [82] in that the proposed control framework satisfies the $\mathcal{C}^1$ continuity, whereas the controller in [182] only meets the continuity conditions but *not* the $\mathcal{C}^1$ continuity, and [82] uses the CLF condition to parameterize the rigid coupling between two subsystems for distributed control of a quadruped without addressing any continuity of the control law. As a brief aside, this dissertation focuses on hierarchical control frameworks for interconnected systems, as will be discussed more in the latter chapters. Although we address the interaction forces between agents at the planning level, it would be an interesting line of research to take an approach similar to that of [82] to decouple collaborative systems through the use of CLFs and implement the control at the low-level only.

The content of this chapter is primarily comprised of material from [43].

2.2 Hybrid Model of Locomotion

The objective of this section is to address hybrid dynamical models of quadrupedal locomotion. We consider floating-based models for general quadrupedal robots whose legs end at point feet. The generalized coordinates of the robot are assumed to be denoted by $q \in \mathcal{Q} \subset \mathbb{R}^{n_q}$, where $\mathcal{Q}$ represents the configuration space for some positive integer n_q representing the dimensions of the generalized coordinates. The state vector can be taken as $x := \mathrm{col}(q, \dot{q}) \in \mathcal{X} \subset \mathbb{R}^n$, in which $\mathcal{X} := \mathrm{T}\mathcal{Q}$ denotes the state space with $n = 2n_q$. In addition, the joint-level torque inputs are shown by $\tau \in \mathcal{T} \subset \mathbb{R}^m$, where $\mathcal{T}$ is a closed and convex admissible set of inputs for some $m < n_q$. The equations of motion can be described by the following ordinary differential equations (ODEs)

$$D(q)\,\ddot{q} + H\,(q, \dot{q}) = B\,\tau + \sum_{\ell \in \mathcal{G}} J_\ell^\top(q)\,\lambda_\ell, \tag{2.1}$$

where $D(q) \in \mathbb{R}^{n_q \times n_q}$ denotes the symmetric and positive definite mass-inertia matrix, $H(q, \dot{q}) \in \mathbb{R}^{n_q}$ represents the Coriolis, centrifugal, and gravitational terms, and $B \in \mathbb{R}^{n_q \times m}$ denotes the input distribution matrix. In addition, $\mathcal{G}$ represents the index set of ground contact points, $J_\ell(q) := \frac{\partial p_\ell}{\partial q}(q) \in \mathbb{R}^{3 \times n_q}$ denotes the Jacobian matrix at the contact point $\ell \in \mathcal{G}$, $p_\ell(q) \in \mathbb{R}^3$ represents the Cartesian coordinates of the contact point, and $\lambda_\ell \in \mathbb{R}^3$ is the corresponding GRF. By defining $\lambda := \mathrm{col}\{\lambda_\ell \,|\, \ell \in \mathcal{G}\}$, the state equation can be expressed as

$$\dot{x} = f(x) + g(x)\,\tau + w(x)\,\lambda \tag{2.2}$$

subject to the holonomic constraints $\ddot{p} = 0$, where $p := \text{col}\{p_\ell \,|\, \ell \in \mathcal{G}\}$ represents the Cartesian coordinates of all contact points, which implies rigid contact and no foot slippage. More formally, we have $\ddot{p}_\ell = J_\ell(q)\,\ddot{q} + \frac{\partial}{\partial q}\left(J_\ell(q)\,\dot{q}\right)\dot{q} = 0$ for all $\ell \in \mathcal{G}$. We remark that this model is valid if $\lambda_\ell \in \mathcal{FC}$ for all $\ell \in \mathcal{G}$, where

$$\mathcal{FC} := \{\text{col}(\lambda_x, \lambda_y, \lambda_z) \,|\, \lambda_z \geq 0, \pm\lambda_x \leq \frac{\mu}{\sqrt{2}}\lambda_z, \pm\lambda_y \leq \frac{\mu}{\sqrt{2}}\lambda_z\}$$

denotes the friction cone for some friction coefficient μ.

Remark 2.1. We remark that one can eliminate the Lagrange multipliers (i.e., GRFs) λ from the ODE (2.2) to satisfy the algebraic holonomic constraints $\ddot{p} = 0$. However, this can result in complicated continuous-time dynamics. In this case, we would need to recompute λ to address the feasibility conditions in the QP of Section 2.3. This motivates us not to eliminate λ. In addition, we have observed that keeping the Lagrange multipliers in the ODE while considering the holonomic constraints for synthesizing the optimal control problem can result in computationally tractable QPs. This will be clarified more in Sections 2.3 and 2.5.

Quadrupedal locomotion can be expressed by multi-domain hybrid systems. Using [40, Theorem 4.3], the stability analysis of periodic orbits for multi-domain hybrid models can be reduced to that of single-domain hybrid models. In this approach, the reset map for the equivalent single-domain hybrid system can be expressed as the composition of the flows of the remaining continuous-time domains and discrete-time transitions in the order they are executed in the multi-domain hybrid systems' cycle. In particular, assuming domain 1 is the main continuous-time domain, the equivalent reset law can be expressed as $\Delta := \Delta_{N\to 1} \circ \mathcal{F}_N \circ \cdots \circ \Delta_{2\to 3} \circ \mathcal{F}_2 \circ \Delta_{1\to 2}$, where N denotes the number of domains and $\mathcal{F}_i$ and $\Delta_{i\to j}$ represent the flow of the continuous-time domain i and the reset law during the discrete-time transition $i \to j$ [33], respectively, for all $i, j \in \{1, \cdots, N\}$ [55, Sec. IV]. In this chapter, we study periodic orbits corresponding to double-domain trotting gaits, which have left-right symmetry. Using [53, Remark 11], one can apply symmetry to the controller of domain 1 to construct the controller for domain 2. Hence, without loss of generality, we will focus on single-domain hybrid models of locomotion, which can be described as follows:

$$\Sigma : \begin{cases} \dot{x} = f(x) + g(x)\,\tau + w(x)\,\lambda, & x \in \mathcal{X} \\ \ddot{p} = 0 & \\ x^+ = \Delta(x^-), & x^- \in \mathcal{X} \cap \mathcal{S}, \end{cases} \tag{2.3}$$

where $\mathcal{S}$ represents the guard of the hybrid system, referred to as the switching manifold. The state solutions of Σ undergo an abrupt change according to the $\mathcal{C}^1$ reset law $x^+ = \Delta(x^-)$ when they hit the guard $\mathcal{S}$.

Assumption 2.2 (Periodic Orbit)**.** There exists a period-one orbit $\mathcal{O}$ for the system Σ that is transversal to the switching manifold $\mathcal{S}$. In particular, $\mathcal{O} := \{\varphi^\star(t) \,|\, 0 \leq t < T\}$ for some periodic state solution $\varphi^\star(t)$ and some fundamental period $T > 0$. Furthermore, the orbit

intersects the switching manifold in exactly one point, i.e., $\{x^\star\} := \overline{\mathcal{O}} \cap \mathcal{S}$ is a singleton, where $\overline{\mathcal{O}}$ denotes the set closure of $\mathcal{O}$.

2.3 QP-Based Nonlinear Controller with Control Lyapunov Function

This section presents a nonlinear control scheme, based on QP, virtual constraints, and CLFs, for whole-body motion control and robust stabilization of the orbit $\mathscr{O}$. The properties of the QP-based nonlinear controller are studied to show that it becomes $\mathscr{C}^1$ on an open neighborhood of the orbit under reasonable sufficient conditions. Furthermore, we consider a set of holonomic virtual constraints as output functions $y := h(x) \in \mathbb{R}^m$ to be imposed by the action of a feedback controller.

Assumption 2.3 (Output Properties)**.** The output function $y(x)$ is assumed to be smooth (i.e., $\mathcal{C}^\infty$) with uniform relative degree 2 [48] with respect to the control input τ in (2.3) on an open neighborhood of the orbit $\mathcal{O}$. In addition, $y(x)$ vanishes on $\mathcal{O}$, that is, $y(x) = 0$ for all $x \in \overline{\mathcal{O}}$.

Differentiating the output function $y(x)$ along the continuous-time dynamics (2.2) and setting the result equal to the desired output dynamics to solve for τ yields

$$\ddot{y} = \mathrm{L}_g\mathrm{L}_f y\,\tau + \mathrm{L}_w\mathrm{L}_f y\,\lambda + \mathrm{L}_f^2 y = -K_P\,y - K_D\,\dot{y} + v, \tag{2.4}$$

where "L" represents the Lie derivative, K_P and K_D are positive definite matrices, and $v \in \mathbb{R}^m$ is an auxiliary input to be discussed later. We remark that in (2.4), $\ddot{y}$ is an affine function of both the inputs τ and the GRFs λ, and $\mathrm{L}_g\mathrm{L}_f y$ and $\mathrm{L}_w\mathrm{L}_f y$ denote the corresponding decoupling matrices. The right-hand side term $-K_P\,y - K_D\,\dot{y} + v$ represents the desired output dynamics that we ultimately want to achieve by solving for (τ, λ). By defining $\eta := \mathrm{col}(y, \dot{y}) \in \mathbb{R}^{2m}$, the output dynamics (2.4) can be written in a compact form as follows:

$$\dot{\eta} = \bar{f}(\eta) + \bar{g}(\eta)\,v := F\,\eta + G\,v, \tag{2.5}$$

where

$$F := \begin{bmatrix} 0 & I \\ -K_P & -K_D \end{bmatrix} \in \mathbb{R}^{2m\times 2m}, \quad G := \begin{bmatrix} 0 \\ I \end{bmatrix} \in \mathbb{R}^{2m\times m}.$$

Since F is Hurwitz, for every positive definite $Q = Q^\top$, there exists a unique and positive definite $P = P^\top$ such that $F^\top P + P\,F = -Q$. A function $V_\varepsilon(\eta)$ is said to be a *rapidly exponentially stabilizing CLF (RES-CLF)* for (2.5) if there are positive scalars $c_1, c_2, c_3 > 0$

such that for all $0 < \varepsilon < 1$ and $\eta \in \mathbb{R}^{2m}$, the following conditions are met

$$\begin{aligned} & c_1\|\eta\|^2 \le V_\varepsilon(\eta) \le \frac{c_2}{\varepsilon^2}\|\eta\|^2 \\ & \inf_v \left\{ \mathrm{L}_{\bar{f}} V_\varepsilon(\eta) + \mathrm{L}_{\bar{g}} V_\varepsilon(\eta)\, v + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \right\} \le 0. \end{aligned} \tag{2.6}$$

Following [41], $V_\varepsilon(\eta) := \eta^\top P_\varepsilon\, \eta$ is an RES-CLF for (2.5) with $P_\varepsilon := \mathrm{diag}(\frac{1}{\varepsilon} I, I)\, P\, \mathrm{diag}(\frac{1}{\varepsilon} I, I)$. More specifically, we can show that the CLF condition can be expressed as the following affine inequality in terms of v

$$\dot{V}_\varepsilon + \frac{c_3}{\varepsilon} V_\varepsilon = \psi_0(x) + \psi_1(x)\, v \le 0, \tag{2.7}$$

where $c_3 := \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}$, $\psi_0(x) := \eta^\top (F^\top P_\varepsilon + P_\varepsilon\, F + \frac{c_3}{\varepsilon}\, P_\varepsilon)\, \eta$, and $\psi_1(x) := 2\eta^\top P_\varepsilon\, G$. From Assumption 2.3, $\eta = 0$ for all $x \in \overline{\mathcal{O}}$, and hence, the inequality (2.7) is reduced to the trivial case of $0\, v \le 0$ on the orbit.

Analogous to (2.4), the algebraic holonomic constraints arising from the stationary contacts of the stance leg ends with the ground can be expressed as

$$\ddot{p} = \mathrm{L}_g \mathrm{L}_f p\, \tau + \mathrm{L}_w \mathrm{L}_f p\, \lambda + \mathrm{L}_f^2 p = 0. \tag{2.8}$$

Assumption 2.4. The contact constraints are *regular* in that the square matrix $\mathrm{L}_w \mathrm{L}_f p(x)$ is full-rank for every $x \in \mathcal{X}$.

We aim to solve for (τ, λ, v) that satisfy the output dynamics (2.4) and the contact condition (2.8) while addressing the CLF condition (2.7) as well as the feasibility constraints $\tau \in \mathcal{T}$ and $\lambda \in \mathcal{FC}$. For this purpose, we set up the following real-time strictly convex QP

$$\begin{aligned} \min_{(\tau,\lambda,v,\delta)} \ & \frac{\gamma_1}{2}\|\tau\|^2 + \frac{\gamma_2}{2}\|\lambda - \lambda_{\mathrm{d}}\|^2 + \frac{\gamma_3}{2}\|v\|^2 + \frac{\gamma_4}{2}\delta^2 \\ \text{s.t.} \quad & \mathrm{L}_g \mathrm{L}_f y\, \tau + \mathrm{L}_w \mathrm{L}_f y\, \lambda + \mathrm{L}_f^2 y = -K_P\, y - K_D\, \dot{y} + v \\ & \mathrm{L}_g \mathrm{L}_f p\, \tau + \mathrm{L}_w \mathrm{L}_f p\, \lambda + \mathrm{L}_f^2 p = 0 \\ & \psi_0 + \psi_1\, v \le \delta \\ & \tau \in \mathcal{T}, \quad \lambda \in \mathcal{FC}, \end{aligned} \tag{2.9}$$

where $\gamma_1, \gamma_2, \gamma_3, \gamma_4 > 0$ are weighting factors. The equality constraints of (2.9) correspond to the I-O linearization (2.4) and rigid contact assumption (2.8). The CLF condition (2.7) is then relaxed by introducing a defect variable $\delta \in \mathbb{R}$. Theorem 2.7 will show that this relaxation would allow the $\mathcal{C}^1$ continuity of the optimal solution of the QP with respect to x in an open neighborhood of the orbit $\mathcal{O}$. The QP considers the feasibility condition of the torque inputs and GRFs as inequality constraints. The cost function finally tries to find the minimum 2-norm (minimum power) torques τ that impose the actual GRFs λ to follow

a desired GRF profile $\lambda_{\mathrm{d}}(x)$ while reducing the magnitude of the defect variable δ and the auxiliary input v. Note that in this work, we heuristically choose $\lambda_{\mathrm{d}} = \mathbf{0}$ to try to minimize the overall GRF. This may be sub-optimal in many cases but serves as a good test scenario when there are optimal GRFs are not provided. The optimal GRFs could be determined using a trajectory planner such as those considered later in this work or could be obtained through offline optimization. The existence and uniqueness of the solution to this strictly convex QP will be shown via Assumption 2.5 and Theorem 2.7.

For future purposes, the optimal solution of the QP (2.9) is denoted by $(\tau^\star(x), \lambda^\star(x), v^\star(x), \delta^\star(x))$ and is parameterized by the state vector x. Furthermore, the closed-loop hybrid system can be expressed as

$$\Sigma^{\mathrm{cl}} : \begin{cases} \dot{x} = f^{\mathrm{cl}}(x), & x \in \mathcal{X} \\ \ddot{p} = 0 & \\ x^+ = \Delta(x^-), & x^- \in \mathcal{X} \cap \mathcal{S}, \end{cases} \tag{2.10}$$

where $f^{\mathrm{cl}}(x) := f(x) + g(x)\,\tau^\star(x) + w(x)\,\lambda(x)$ is the closed-loop vector field. We remark that $\lambda(x) = \lambda^\star(x)$ as from (2.8), $\lambda(x)$ can be uniquely computed based on $\tau^\star(x)$.

2.3.1 Continuous Differentiability of the Feedback Controller

The QP in (2.9) can be expressed in a compact form as the following parameterized optimization problem

$$\mathcal{P}(x) : \begin{cases} \min\limits_{\xi} & J(\xi, x) \\ \text{s.t.} & \rho_i(\xi, x) = 0, \;\; i \in \mathcal{I}_{\mathrm{eq}} := \{1, \cdots, n_{\mathrm{eq}}\} \\ & \omega_j(\xi, x) \leq 0, \;\; j \in \mathcal{I}_{\mathrm{ineq}} := \{1, \cdots, n_{\mathrm{ineq}}\}, \end{cases}$$

where $\xi := \mathrm{col}(\tau, \lambda, v, \delta)$ represents the decision variables to be determined. The Lagrangian for $\mathcal{P}(x)$ is defined as

$$\mathcal{L}(\xi, \alpha, \beta, x) := J(\xi, x) + \sum_{i \in \mathcal{I}_{\mathrm{eq}}} \alpha_i\, \rho_i(\xi, x) + \sum_{j \in \mathcal{I}_{\mathrm{ineq}}} \beta_j\, \omega_j(\xi, x),$$

where α and β are the Lagrange multipliers corresponding to the equality and inequality constraints, respectively. A point $(\xi^\star, \alpha^\star, \beta^\star)$ satisfies the *Karush-Kuhn-Tucker (KKT) conditions* for $\mathcal{P}(x_0)$ if 1) $\frac{\partial \mathcal{L}}{\partial \xi}(\xi^\star, \alpha^\star, \beta^\star, x_0) = 0$, 2) all equality constraints are met at $(\xi, x) = (\xi^\star, x_0)$, 3) all inequality constraints are satisfied at $(\xi, x) = (\xi^\star, x_0)$, and 4) $\beta_j\, \omega_j(\xi^\star, x_0) = 0$ with $\beta_j \geq 0$ for all $j \in \mathcal{I}_{\mathrm{ineq}}$ (complementary slackness). A point $(\xi^\star, \alpha^\star, \beta^\star)$ satisfies *strict complementary slackness* if there is not any $j \in \mathcal{I}_{\mathrm{ineq}}$ for which both $\beta_j = 0$ and $\omega_j(\xi^\star, x_0) = 0$. A point $(\xi^\star, \alpha^\star, \beta^\star)$ is said to be *regular* if the gradients of the active constraints of $\mathcal{P}(x_0)$ are linearly independent.

The point $(\xi^\star, \alpha^\star, \beta^\star)$ satisfies the *second-order sufficient conditions* of the QP $\mathcal{P}(x_0)$ if a) the KKT conditions are met, and b) the Hessian matrix meets the condition $z^\top \frac{\partial^2 \mathcal{L}}{\partial \xi^2}(\xi^\star, \alpha^\star, \beta^\star, x_0)\, z > 0$ for all $z \neq 0$ such that

(1) $z^\top \frac{\partial \rho_i}{\partial \xi}(\xi^\star, x_0) = 0$ for all $i \in \mathcal{I}_{\text{eq}}$,

(2) $z^\top \frac{\partial \omega_j}{\partial \xi}(\xi^\star, x_0) = 0$ for all $j \in \mathcal{I}_{\text{ineq}}$ where $\beta_j^\star > 0$,

(3) $z^\top \frac{\partial \omega_j}{\partial \xi}(\xi^\star, x_0) \leq 0$ for all $j \in \mathcal{I}_{\text{ineq}}$ where $\beta_j^\star = 0$.

Assumption 2.5. *(Optimality on the Periodic Orbit):* We suppose that, for all x_0 on the orbit $\overline{\mathcal{O}}$, the QP $\mathcal{P}(x_0)$ is feasible and there exists a point $(\xi^\star(x_0), \alpha^\star(x_0), \beta^\star(x_0))$ that satisfies the second-order sufficient conditions. The point $(\xi^\star(x_0), \alpha^\star(x_0), \beta^\star(x_0))$ also satisfies the strict complementary slackness for $\mathcal{P}(x_0)$. We further suppose that the optimal control and GRFs take values in the interior of the sets $\mathcal{T}$ and $\mathcal{FC}$, that is, $\tau^\star(x_0) \in \text{int}(\mathcal{T})$ and $\lambda^\star(x_0) \in \text{int}(\mathcal{FC})$, where "int" represents the interior of a set.

Remark 2.6. Assumption 2.5 is *not* restrictive and states that the QP has a feasible solution that satisfies the second-order sufficient conditions and complementary slackness for every point on the desired orbit, which follows simply from the strict convexity of the problem. It also states that the torques and GRFs corresponding to the desired trajectory remain in the interior of the feasible sets, which can practically be met during trajectory optimization of the desired periodic orbit $\mathcal{O}$. In particular, the trajectory optimization problem for generating $\mathcal{O}$ can be constrained by a *conservative* subset of the feasible sets $\mathcal{T}$ and $\mathcal{FC}$ such that the desired orbit remains in the interior of the actual feasible sets $\mathcal{T}$ and $\mathcal{FC}$ at all times during locomotion.

Theorem 2.7. *(Existence, Uniqueness, and $\mathcal{C}^1$ Continuity of the Optimal Solution): Under Assumptions 2.2-2.5, there exist an open neighborhood of the periodic orbit $\mathcal{O}$, denoted by $\mathcal{N}(\mathcal{O})$, and a continuously-differentiable function $\xi^\star(x) := col(\tau^\star(x), \lambda^\star(x), v^\star(x), \delta^\star(x))$, such that for all $x \in \mathcal{N}(\mathcal{O})$, $\xi^\star(x)$ is an isolated optimal solution of the QP $\mathcal{P}(x)$.*

Proof. Since $f(x)$, $g(x)$, $w(x)$, and $y(x)$ are smooth (i.e., $\mathcal{C}^\infty$), the cost function $J(\xi, x)$ and constraints $\rho(\xi, x)$ and $\omega(\xi, x)$ are smooth in x on an open neighborhood of the orbit. In addition, $J(\xi, x)$ and constraints $\rho(\xi, x)$ and $\omega(\xi, x)$ are smooth in ξ. We next show that for all $x_0 \in \overline{\mathcal{O}}$, the optimal solution of $\mathcal{P}(x_0)$, denoted by $(\xi^\star, \alpha^\star, \beta^\star)$, satisfies the regularity condition. From Assumption 2.3, $\ddot{y} = 0$ on the orbit. Hence, according to the output dynamics (2.4), the optimal v value must be zero, i.e., $v^\star = 0$. Since $\psi_0 = 0$ and $\psi_1 = 0$ on the orbit, the relaxed-CLF condition in (2.9) is reduced to $0 \leq \delta$. According to the positive term $\frac{\gamma_4}{2}\delta^2$ in the cost function, we can conclude that $\delta^\star = 0$. Hence, the relaxed CLF condition, which is expressed as an inequality constraint, is indeed active on the orbit. From Assumption 2.5, we also have that $\tau^\star \in \text{int}(\mathcal{T})$ and $\lambda^\star \in \text{int}(\mathcal{FC})$. Consequently, the feasibility constraints $\tau \in \mathcal{T}$ and $\lambda \in \mathcal{FC}$ of the QP (2.9) are inactive on the orbit. We now

study the rank of the gradients of the active constraints with respect to $\xi = \mathrm{col}(\tau, \lambda, v, \delta)$ that is reduced to the following matrix on the orbit

$$\begin{bmatrix} \mathrm{L}_g\mathrm{L}_f y(x_0) & \mathrm{L}_w\mathrm{L}_f y(x_0) & -I & 0 \\ \mathrm{L}_g\mathrm{L}_f p(x_0) & \mathrm{L}_w\mathrm{L}_f p(x_0) & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{2.11}$$

By Assumption 2.4, $\mathrm{L}_w\mathrm{L}_f p(x_0)$ is full-rank, and hence, the gradient matrix in (2.11) has full row rank for every $x_0 \in \overline{\mathcal{O}}$. Thus, all sufficient conditions of Fiacco's Theorem (see [185, Theorem 2.1] or [182, Theorem 1]) are met, resulting in the existence, uniqueness, and $\mathcal{C}^1$ continuity of optimal solutions of the QP on an open neighborhood of the orbit $\mathcal{O}$. □

2.4 Robust Stability Analysis

The objective of this section is to address the robust stabilization problem of the periodic orbit $\mathcal{O}$ based on the Poincaré sections analysis and ISS. We consider the closed-loop hybrid model (2.10) subject to external disturbances during the continuous-time domain as follows:

$$\dot{x} = f^{\mathrm{cl}}(x) + a(x)\, d, \tag{2.12}$$

where $a(x)$ is a smooth function, and d is an external wrench (i.e., disturbance) defined by a finite-dimensional set of parameters [183, Sec. II.C]. Typical examples include constant disturbance inputs or splines whose parameters change from one domain to another. We suppose that $d_k \in \mathcal{D}$ represents the parameterization of the disturbance during the k-th continuous-time domain, where $\mathcal{D}$ is a domain containing the origin. The evolution of the perturbed hybrid system on the Poincaré section $\mathcal{S}$ can then be described by the following discrete-time dynamics

$$x_{k+1} = \mathcal{R}(x_k, d_k), \quad k = 0, 1, \cdots, \tag{2.13}$$

where $\mathcal{R} : \mathcal{S} \times \mathcal{D} \to \mathcal{S}$ represents the Poincaré return map parameterized by the disturbance d_k. To study the properties of the Poincaré map, we make the following assumption.

Assumption 2.8. We suppose that for all $x \in \overline{\mathcal{O}}$, the matrix

$$T(x) := \mathrm{L}_g\mathrm{L}_f y - \mathrm{L}_w\mathrm{L}_f y \, (\mathrm{L}_w\mathrm{L}_f p)^{-1} \, \mathrm{L}_g\mathrm{L}_f p \in \mathbb{R}^{m \times m}$$

is full-rank.

Remark 2.9. Assumption 2.8 is *not* restrictive and is met inherently if the system is not overactuated. In this work, we consider a trot gait that does not have an overactuated continuous-time domain, so this assumption is satisfied. In the proof of Theorem 2.11, we will show that this ensures the uniqueness of the torques corresponding to the periodic gait.

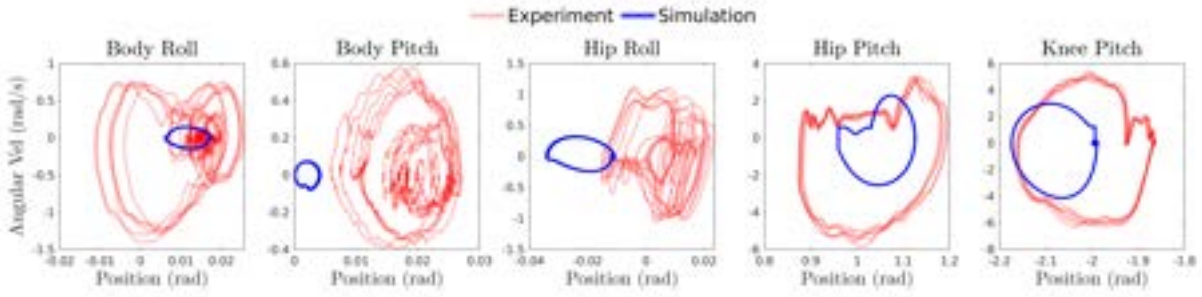

Figure 2.1: Simulated and experimental data for a forward trot subject to a payload of 4.54 (kg). The joint phase plots are shown for the front right leg.

Definition 2.10. A fixed point $x^\star$ is said to be locally ISS (LISS) for (2.13), if there exists $\epsilon > 0$, a class $\mathcal{KL}$ function ϱ, and a class $\mathcal{K}$ function ϖ such that

$$\|x_k - x^\star\| \leq \varrho(\|x_0 - x^\star\|, k) + \varpi\left(\|d\|_{l_\infty}\right), \ \forall k = 0, 1, \cdots,$$

for all $x_0 \in \mathcal{S} \cap \mathcal{B}_\epsilon(x^\star)$ and $d \in \mathcal{B}_\epsilon(0)$, where $\mathcal{B}_\epsilon(x^\star)$ and $\mathcal{B}_\epsilon(0)$ are open ϵ-neighborhood balls around $x^\star$ and 0, respectively, and $\|d\|_{l_\infty}$ represents the l_∞-norm.

We are now in a position to present the following theorem to investigate the existence of a fixed point and its LISS property for the Poincaré return map.

Theorem 2.11. *(Invariance and Robust Stability): Under Assumptions 2.2-2.8, the following statements hold.*

- ***(1)*** *The orbit $\mathcal{O}$ is invariant under the flow of the closed-loop hybrid system in the absence of the disturbance d. In particular, $x^\star$ is a fixed point for the Poincaré map in the absence of d, that is $\mathcal{R}(x^\star, 0) = x^\star$.*
- ***(2)*** *If the eigenvalues of $\Pi_0 := \frac{\partial \mathcal{R}}{\partial x}(x^\star, 0)$ are strictly inside the unit circle, then $x^\star$ is LISS for (2.13).*

Proof. Part (1): From Assumptions 2.3 and 2.5 and the proof of Theorem 2.7, for every $x \in \overline{\mathcal{O}}$, the QP $\mathcal{P}(x)$ is feasible and the optimal v value is zero (i.e., $v^\star = 0$). Hence, the equality constraints are reduced to $\mathrm{L}_g\mathrm{L}_f y\,\tau + \mathrm{L}_w\mathrm{L}_f y\,\lambda + \mathrm{L}_f^2 y = 0$ and $\mathrm{L}_g\mathrm{L}_f p\,\tau + \mathrm{L}_w\mathrm{L}_f p\,\lambda + \mathrm{L}_f^2 p = 0$. Eliminating the GRFs from these equations, we can conclude that

$$T(x)\,\tau + \mathrm{L}_f^2 y - \mathrm{L}_w\mathrm{L}_f y\left(\mathrm{L}_w\mathrm{L}_f p\right)^{-1}\mathrm{L}_f^2 p = 0. \tag{2.14}$$

This, together with Assumption 2.8, implies that τ is a unique solution for this set of equations which coincides with the open-loop control input that generates the orbit. Hence, $\mathcal{O}$ is invariant under the flow of the closed-loop hybrid dynamics.

Figure 2.2: This experiment displays A1 trotting under the influence of both a 4.54 (kg) payload (36% of the robot's weight) and push disturbances. The quadruped was able to resist these uncertainties and continue trotting. Videos of the experiments are available online [9].

Part (2): Unlike [183], the closed-loop vector field is $\mathcal{C}^1$, but *not* twice continuously differentiable (i.e., $\mathcal{C}^2$). Theorem 2.7 together with the transversality condition in Assumption 2.2 implies that $\mathcal{R}(x,d)$ is $\mathcal{C}^1$ with respect to (x,d) on an open neighborhood of $(x^\star,0)$. Since Π_0 is a Hurwitz matrix, for every $Q_0 = Q_0^\top > 0$, there is a unique $P_0 = P_0^\top > 0$ such that the discrete-time Lyapunov equation $\Pi_0^\top P_0 \Pi_0 - P_0 = -Q_0$ is satisfied. This shows zero-input exponential stability and thereby zero-input asymptotic stability of the fixed point $x^\star$ for the Poincaré return map. We can then conclude the desired local ISS property holds by invoking [186]. More formally, we can choose the Lyapunov function $W(x) := \delta x^\top P_0 \, \delta x$, where $\delta x := x - x^\star$. From [19, Lemma 2], there are $\epsilon, \zeta_0, \sigma_0 > 0$ such that for all $x \in \mathcal{S} \cap \mathcal{B}_\epsilon(x^\star)$ and all $d \in \mathcal{B}_\epsilon(0) \subset \mathcal{D}$,

$$\Delta W := W(\mathcal{R}(x,d)) - W(x) \leq -\zeta_0 \, \|\delta x\|^2 + \sigma_0 \|d\|^2,$$

where $\mathscr{B}_\epsilon(x^\star)$ and $\mathscr{B}_\epsilon(0)$ are open ϵ-neighborhood balls around $x^\star$ and 0, respectively. Since we have

$$\lambda_{\min}(P_0) \, \|\delta x\|^2 \leq W(x) \leq \lambda_{\max}(P_0) \, \|\delta x\|^2,$$

we may conclude that $W_{k+1} \leq \nu \, W_k + \sigma_0 \|d_k\|^2$, where $\nu := 1 - \frac{\zeta_0}{\lambda_{\max}(P_0)} < 1$. For $d = 0$, this inequality is reduced to $W_{k+1} \leq \nu \, W_k$, and hence, $\nu \in [0,1)$. We can show that

$$W_k \leq \nu^k \, W_0 + \sigma_0 \sum_{j=0}^{k-1} \nu^{k-1-j} \|d_j\|^2 \leq \nu^k \, W_0 + \frac{\sigma_0}{1-\nu} d_{\max}^2,$$

where $d_{\max} := \sup_{k \geq 0} \|d_k\|$. This latter inequality together with the property $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ results in

$$\|x_k - x^\star\| \leq \sqrt{\frac{\lambda_{\max}(P_0)}{\lambda_{\min}(P_0)}} \, \|x_0 - x^\star\| \, (\sqrt{\nu})^k + \sqrt{\frac{\sigma_0}{\lambda_{\min}(P_0) \, (1-\nu)}} \, d_{\max}, \tag{2.15}$$

for all $k = 1, 2, \cdots$ which completes the proof. □

2.5 Numerical and Experimental Results

This section aims to numerically and experimentally evaluate the effectiveness of the proposed nonlinear control scheme. We consider a full-order dynamical model of the quadrupedal robot A1 made by Unitree. The floating-based model of the mechanical system consists of 18 DOFs. Six DOFs are unactuated and describe the absolute position and orientation of the robot. The remaining 12 DOFs are the actuated joints of the legs. More specifically, each leg has a 2 DOF hip joint plus a 1 DOF knee joint. The robot weighs approximately 12.45 (kg) and stands up to 0.28 (m) off the ground. In this chapter, we consider a heuristic and symmetric periodic orbit $\mathcal{O}$ for trotting at 0.1 (m/s). We remark that the orbit can also be designed via trajectory optimization techniques. The orbit satisfies Assumptions 2.5 and 2.8. We then consider 12 virtual constraints to stabilize the orbit according to Assumption 2.3. The first three components are defined in the Cartesian space to track the desired trajectories for the geometric center of the robot. The next three components are defined to regulate the orientation of the torso. The remaining components are defined in the Cartesian space to impose the swing leg ends to follow the desired trajectories starting from the previous footholds and ending at the upcoming ones.

The proposed QP-based controller in (2.9) is solved using qpSWIFT [187] at 1kHz on an external laptop with an Intel® Core™ i7-1185G7 running at 3.00 GHz and 16 GB of RAM. Under nominal conditions, the computation time is 0.22 (ms) on average over the course of one domain. The QP uses $\gamma_n = \{1, 0.1, 1e6, 1e8\}$ for the weights and assumes a coefficient of friction of $\mu = 0.7$, which results in stable locomotion.

The proposed controller was first simulated in RaiSim [188], which assumes a rigid contact model (i.e., utilizing (2.8)). Under nominal conditions, the controller results in stable trotting. This is further examined by subjecting the robot to push and payload disturbances that are *unknown* to the controller. Similarly, hardware experiments were performed under several disturbance conditions. The phase plots in Fig. 2.1 display the simulated and experimental results of a trot gait subject to a constant payload with a mass of 4.54 (kg), which is 36% of the total body mass. The gap between simulated and experimental results can be attributed to poorly modeled system dynamics, compliant feet, lack of rigorous contact and state estimation, and differences in the position and mass of the payload. In light of these potential shortcomings, the robot is able to remain stable without knowledge of the payload. In addition to adequately handling this unmodeled payload, the robot was further able to robustly resist push disturbances during experiments without becoming unstable. Snapshots of the experiment involving both a payload and push disturbances can be found in Fig. 2.2 and the corresponding CLF may be found in Fig. 2.3. Even under these disturbances, the theoretical derivative of the CLF remains negative for nearly the entire trial and becomes positive for only brief moments (e.g., δ remains small). It should be noted that the derivative provided depends on the solution to the QP. Therefore, the plot demonstrates the theoretical value of the derivative at each time step subject to the optimal value of v from (2.9). It can be observed that the CLF spikes during the pushes and slowly decreases as the

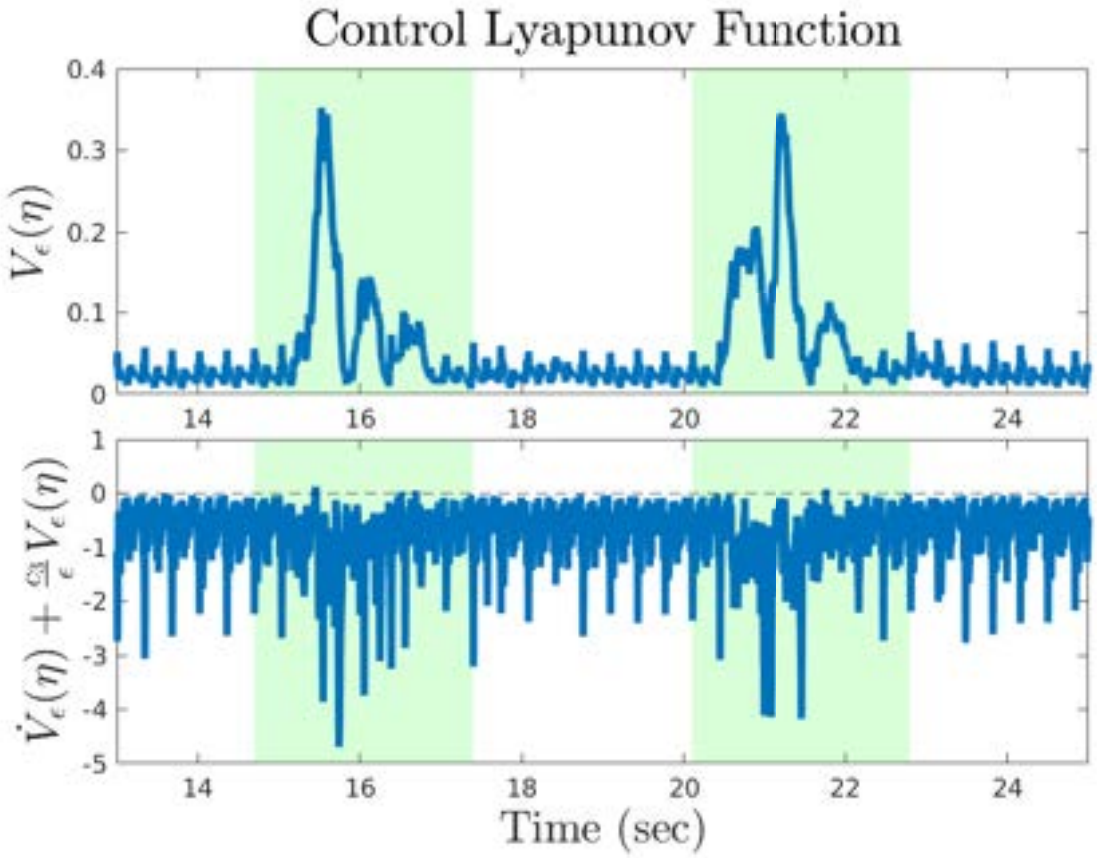

Figure 2.3: CLF corresponding to the experiment in Fig. 2.2. The highlighted portions indicate the pushes and corresponding recovery of the robot. The top plot corresponds to the *actual* Lyapunov function calculated using the current state. The bottom plot provides the *theoretical* derivative calculated from the current state and the optimal solution determined by the QP. In particular, the derivative depends on the decision variable v from (2.9).

robot continues to step forward. Due to the hybrid nature of locomotion, convergence back to the orbit is subject to constantly changing contact domains, leading to slow recovery. In addition, in this work, we aim to track a heuristic trajectory and command a constant lateral position for the robot. This can also lead to oscillatory motions. In particular, if the robot is perturbed significantly from the desired position, it will aggressively attempt to move back to the desired position, often resulting in overshoot and a longer settling time, as is evident from the experimental results. This can be alleviated by providing an optimal trajectory that is determined in an online manner. This topic will be explored more thoroughly in the following chapters. videos of the experiments are available online [9].

2.6 Summary

This chapter presented a nonlinear control scheme, based on virtual constraints, CLFs, and QPs, to robustly stabilize periodic orbits for hybrid dynamical models of quadrupedal locomotion. The first theoretical contribution of this chapter established sufficient conditions such that this QP-based controller is continuously differentiable on an open neighborhood of the orbit. We subsequently showed the invariance of the orbit and its robust stability via the Poincaré sections analysis and ISS. The effectiveness of the proposed controller was verified both numerically and experimentally on the A1 quadrupedal robot. The full-order and nonlinear controller was implemented on the robot as a model-based CLF-QP in real-time. The robust stability of trotting gaits against external disturbances and uncertainties arising from unknown payloads was demonstrated in practice. Future research will investigate the robustness of the gaits on rough terrains. We will also explore the integration of this control scheme with higher-level and MPC-based planning algorithms.

Part II

Data-Driven Trajectory Planning

Chapter 3

Data-Driven Reduced-Order Models for Single Agent Locomotion

3.1 Introduction and Motivation

As has been discussed throughout this work, many current state-of-the-art approaches for planning or controlling legged robots rely on a reduced-order (i.e., template) model of the robot [95]. 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. However, choosing a reduced-order model that properly encapsulates the dominant dynamics can be a difficult task. 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. Although the original theory considered in this work does not directly apply to complex hybrid nonlinear systems, recent works have provided theoretical extensions to certain classes of nonlinear systems [178], implementations for stochastic and nonlinear systems [172, 173, 174], and linear parameter varying systems [179]. 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.

The *overarching goal* of this chapter 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 chapter'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. 3.1). 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 quadrupeds. 3) The optimal trajectories are then passed to a low-level nonlinear controller based on virtual constraints for whole-body motion control

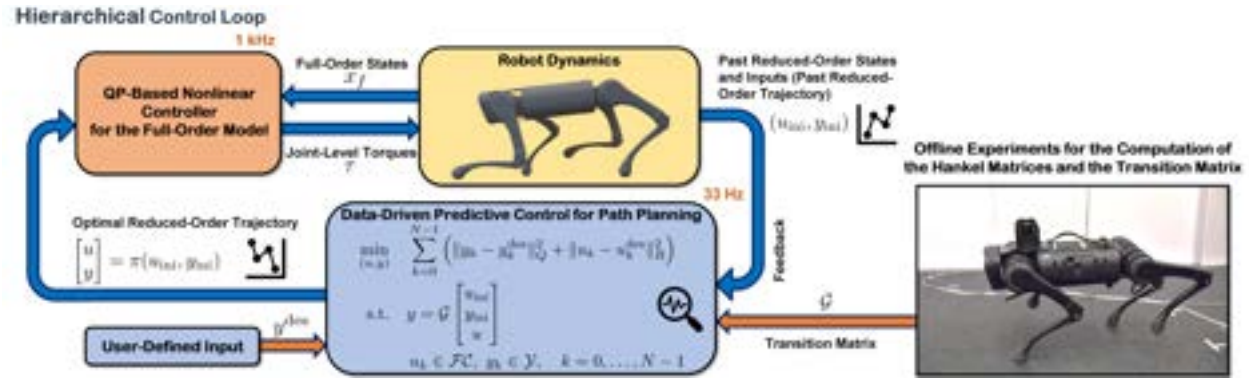

Figure 3.1: Overview of the proposed hierarchical control algorithm. At the high level, the data-driven predictive control generates optimal trajectories for trajectory planning of the quadrupedal robot. The optimal trajectories are then passed to a low-level and QP-based nonlinear controller for the whole-body motion control. The data-driven transition matrix is computed based on a set of offline experiments.

(see e.g., Chapter 2). 4) Experimental validation of the proposed layered control approach is provided on the 18-DOF quadrupedal robot A1 for a walk and trot gait. We further provide experiments that consider a variety of different gait parameters for trotting to explore the robustness of the model to different parameters without collecting new data. The experimental results also show robust locomotion of the A1 robot on unknown terrains and in the presence of disturbances. The majority of the contents of this chapter are taken from our previous work [177] with some extensions.

3.2 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

$$\begin{aligned} x_{k+1} &= A\, x_k + B\, u_k \\ y_k &= C\, x_k + D\, u_k, \end{aligned} \tag{3.1}$$

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., $u^{\mathrm{d}} := \mathrm{col}(u_0^{\mathrm{d}}, \ldots, u_{T-1}^{\mathrm{d}})$. In our notation, "col" represents the column operator. As will be discussed shortly, L represents the sum of the prediction and estimation horizons and is used primarily for notational compactness. Using this trajectory, one can construct the corresponding Hankel matrix [172] as follows:

$$\mathscr{H}_L(u^{\mathrm{d}}) := \begin{bmatrix} u_0^{\mathrm{d}} & u_1^{\mathrm{d}} & \cdots & u_{T-L}^{\mathrm{d}} \\ u_1^{\mathrm{d}} & u_2^{\mathrm{d}} & \cdots & u_{T-L+1}^{\mathrm{d}} \\ \vdots & \vdots & \ddots & \vdots \\ u_{L-1}^{\mathrm{d}} & u_L^{\mathrm{d}} & \cdots & u_{T-1}^{\mathrm{d}} \end{bmatrix} \in \mathbb{R}^{mL \times (T-L+1)}. \tag{3.2}$$

Definition 3.1 ([171])**.** 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 3.2 ([171])**.** The sequence $\{(u_k^{\mathrm{d}}, y_k^{\mathrm{d}})\}_{k=0}^{T-1}$ is said to be a *trajectory* of the LTI system (3.1) 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 (3.1).

Using Definitions 3.1 and 3.2, 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 3.3. *[168, 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*

$$\begin{bmatrix} \mathscr{H}_L(u^d) \\ \mathscr{H}_L(y^d) \end{bmatrix} g = \begin{bmatrix} \bar{u} \\ \bar{y} \end{bmatrix}. \tag{3.3}$$

Theorem 3.3 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 3.3.

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 $L = T_{\mathrm{ini}} + N$. 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 (3.3). 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 (3.3) as follows:

$$\mathscr{H}_L(u^{\mathrm{d}}) = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \quad \mathscr{H}_L(y^{\mathrm{d}}) = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}, \tag{3.4}$$

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 $T \geq (m+1)(T_{\text{ini}} + N + n) - 1$.

3.3 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.

3.3.1 Data-Driven Predictive Control

This section outlines an approach to address predictive control without a physics-based model. In particular, we consider the regularized DeePC methodology provided in [172, 173] as follows:

$$\begin{aligned} \min_{(g,u,y,\sigma)} \quad & \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\text{des}}\|_Q^2 + \|u_k\|_R^2 \right) + \lambda_g \|g\|^2 + \lambda_\sigma \|\sigma\|^2 \\ \text{s.t.} \quad & \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g + \begin{bmatrix} 0 \\ \sigma \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix} \\ & u_k \in \mathcal{U}, \quad y_k \in \mathcal{Y}, \quad k = 0, \ldots, N-1 \end{aligned} \tag{3.5}$$

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 (3.5) to remain feasible in the wake of noisy measurements. If no noise is present, then Theorem 3.3 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 (3.5). From another perspective, the estimation portion may be viewed as a manner in which we can ensure that the future trajectories are dynamically consistent with the past trajectory. Put simply, we must ensure that the previous and future trajectories align sufficiently well. 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 minimum realization matrices in (3.1) 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 (3.5) 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 (3.5), 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 (3.5). Analogous to [173], finding an approximation of g reduces to the following offline optimization problem

$$\begin{aligned} &\min_{g} \ \|g\|^2 \\ &\text{s.t.} \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}. \end{aligned} \tag{3.6}$$

The closed-form expression of (3.6) can be described by

$$g = \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^{\dagger} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}, \tag{3.7}$$

where $(\cdot)^{\dagger}$ represents the pseudo inverse. Using the fact that $y = Y_f\, g$ from (3.5), we have

$$y = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}, \quad \mathcal{G} := Y_f \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^{\dagger}, \tag{3.8}$$

where $\mathcal{G}$ denotes the *data-driven state transition matrix over N-steps*. Using (3.8), we are now in position to present the general form of a computationally tractable predictive

controller based on data for trajectory planning

$$\begin{aligned} \min_{(u,y)} \quad & \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\text{des}}\|_Q^2 + \|u_k\|_R^2 \right) \\ \text{s.t.} \quad & y = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} \\ & u_k \in \mathcal{U}, \ y_k \in \mathcal{Y}, \qquad k = 0, \dots, N-1. \end{aligned} \tag{3.9}$$

Remark 3.4. In this formulation, there is no longer a defect variable to mitigate the influence noise has on the prediction. Practically speaking, using large amounts of data considerably reduces the impact of noise to the point where the defect variable is no longer necessary. However, in the event that there are still issues with noise, one additional possibility would be to consider using a Page matrix as opposed to a Hankel matrix, which has superior noise properties but requires larger amounts of data [174].

Remark 3.5. Careful consideration is required when performing this approximation. In particular, we remark that g in (3.5) 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 3.3, if properly constructed, all possible trajectories of (3.1) 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.

3.3.2 Trajectory Planning for Quadrupedal Robots

In this section, we discuss the application of the data-driven predictive control of (3.9) to the real-time planning of quadrupeds and draw relations to the common SRB template model. The nonlinear SRB model is described by [4, 89, 90]

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} p_c \\ \dot{p}_c \\ R \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{p}_c \\ \frac{1}{m^{\text{net}}} f^{\text{net}} - g_0 \, e_z \\ R \, \hat{\omega} \\ I_r^{-1} (R^\top \tau^{\text{net}} - \hat{\omega} \, I_r \, \omega) \end{bmatrix}, \tag{3.10}$$

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 (3.10) can be described by

$$\begin{bmatrix} f^{\text{net}} \\ \tau^{\text{net}} \end{bmatrix} = \sum_{\ell \in \mathcal{C}} \begin{bmatrix} I_{3\times 3} \\ \hat{d}_\ell \end{bmatrix} f_\ell, \tag{3.11}$$

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 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 [87]. It should be noted that, even when using longer horizons, it is still possible to only consider the current domain when the MPC is updated at a relatively high frequency, i.e., 100+ (Hz).

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 (3.10), 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 3.5.

Remark 3.6. 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 (3.11). 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, and this does not seem to pose an issue at this time. We further remark that, although we have chosen to mimic the SRB model in this work, one could similarly consider using the states and constraints found in other template models, such as the LIP model, for use in this data-driven scheme.

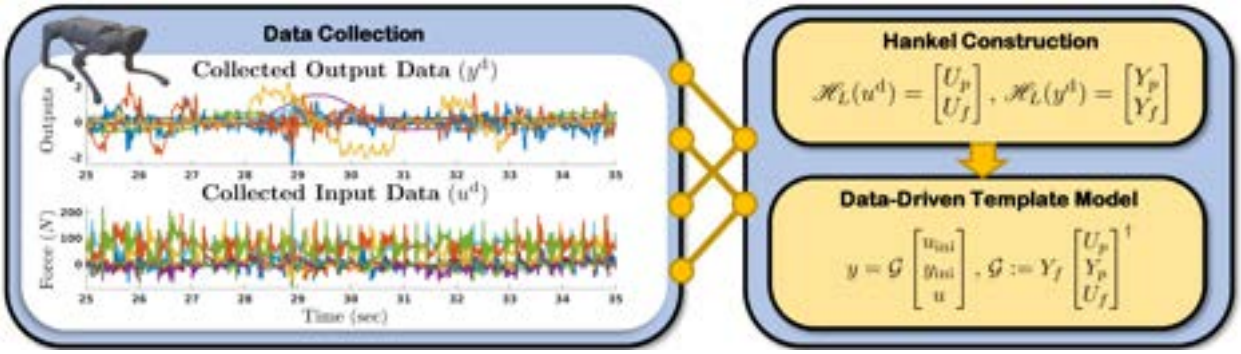

Figure 3.2: Overview of the process used to construct the data-driven template mode. The data is collected by directly using the QP-based low-level controller (3.18), and that data is then used to construct a template model on which a predictive trajectory planner can be based.

The data-driven trajectory planner is then defined by

$$\begin{aligned} \min_{(u,y)} \quad & \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\text{des}}\|_Q^2 + \|u_k - u_k^{\text{des}}\|_R^2 \right) \\ \text{s.t.} \quad & y = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} \\ & u_k \in \mathcal{FC}, \; y_k \in \mathcal{Y}, \quad k \in 0, \ldots, N-1, \end{aligned} \tag{3.12}$$

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 by

$$u_{k,\ell}^{\text{des}} := \begin{cases} \text{col}\left(0, 0, \dfrac{m^{\text{net}} g_0}{N_{c,k}}\right), & \ell \in \mathcal{C}_k \\ \text{col}(0,0,0), & \text{Otherwise.} \end{cases} \tag{3.13}$$

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

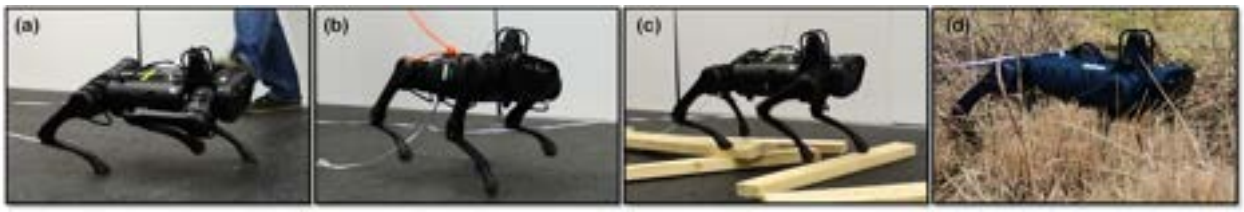

Figure 3.3: Snapshots from experiments with the proposed hierarchical control algorithm: (a) external push disturbances, (b) external tethered pull disturbances, (c) unknown rough terrain covered with wooden blocks, and (d) unstructured and unknown outdoor environment. Videos of these experiments can be found online [10].

a similar manner such that the forces on anticipated swing legs are restricted to zero, while the anticipated 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 or simply weigh the robot.

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 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 [4, 89, 90].

3.4 Nonlinear Low-Level Controller

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

3.4.1 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 $x_f := \text{col}(q, \dot{q}) \in \mathcal{X} \subset \mathbb{R}^{2n_q}$ 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

$$D(q)\,\ddot{q} + H(q, \dot{q}) = \Upsilon\,\tau + J^\top(q)\,f, \tag{3.14}$$

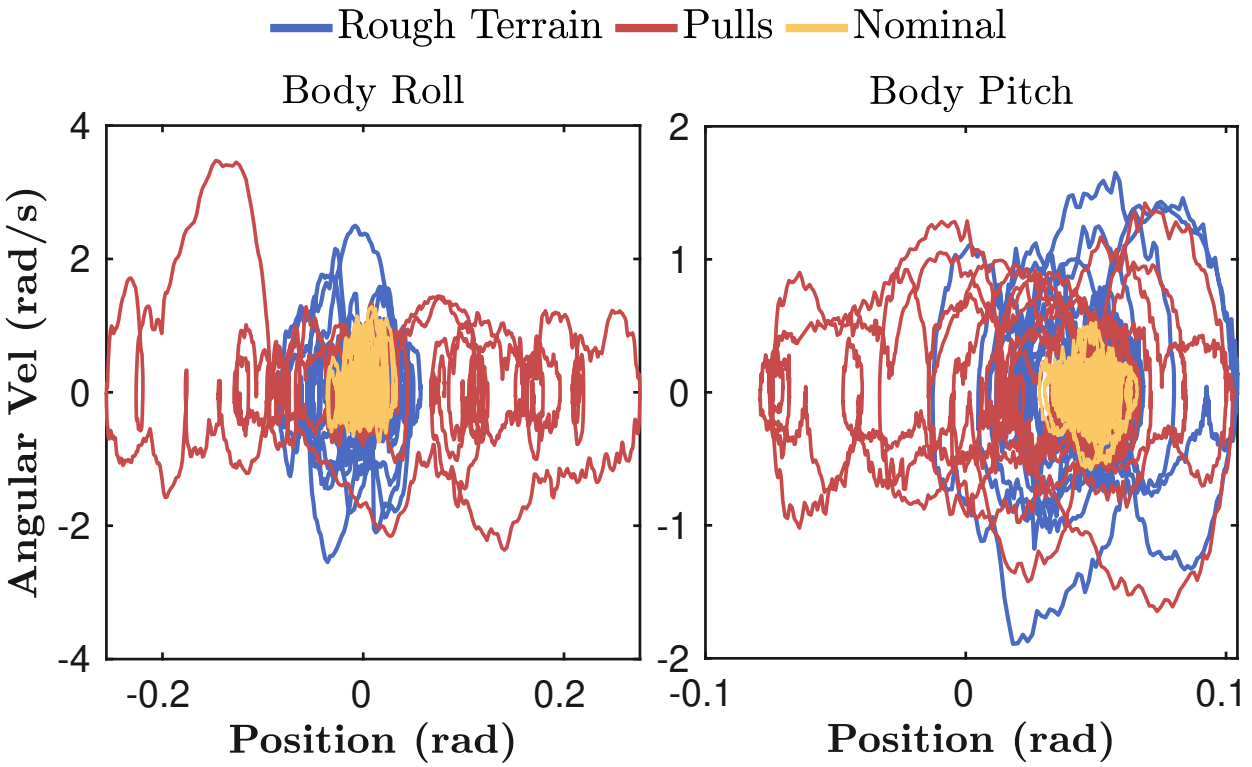

Figure 3.4: Phase portraits of the robot's body orientation (i.e., roll and pitch) during different experiments. The quadruped is able to robustly trot over flat ground (nominal), unknown rough terrain covered with wooden blocks, and subject to external disturbances (pulls). For each experiment, the 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.

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 $f := \text{col}\{f_\ell | \, \ell \in \mathcal{C}\}$ represents the vector GRFs of the contacting leg ends. We further impose the holonomic constraint $\ddot{r} = 0$ on (3.14), where $r := \text{col}\{p_\ell | \, \ell \in \mathcal{C}\}$ represents the position of the contacting leg ends with the environment. This constraint implies rigid contact with the ground and is valid if $f_\ell \in \mathcal{FC}$, $\forall \ell \in \mathcal{C}$.

3.4.2 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* [40] as

$$h(x_f, t) := h_0(q) - h^{\text{des}}(t), \tag{3.15}$$

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 [48]. The term $h_0(q)$ denotes the variables that we are interested in controlling, and $h^{\text{des}}(t)$ 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 [189, Eq. (4), pp. 46]. Differentiating $h(x_f, t)$ twice along the dynamics (3.14), we have

$$\ddot{h} = \Theta_1(x_f)\,\tau + \Theta_2(x_f)\,f + \theta(x_f) = -K_P\,h - K_D\,\dot{h} + \delta, \tag{3.16}$$

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 [103, Appendix A] 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:

$$\ddot{r} = \Phi_1(x_f)\,\tau + \Phi_2(x_f)\,f + \phi(x_f) = 0, \tag{3.17}$$

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 [43]

$$\begin{aligned}
\min_{(\tau,f,\delta)} \; & \frac{\gamma_1}{2}\|\tau\|^2 + \frac{\gamma_2}{2}\|f - f^{\text{des}}\|^2 + \frac{\gamma_3}{2}\|\delta\|^2 \\
\text{s.t.} \; & \Theta_1(x_f)\,\tau + \Theta_2(x_f)\,f + \theta(x_f) = -K_P\,h - K_D\,\dot{h} + \delta \\
& \Phi_1(x_f)\,\tau + \Phi_2(x_f)\,f + \phi(x_f) = 0 \\
& \tau \in \mathcal{T}, \quad f_\ell \in \mathcal{FC}, \quad \forall \ell \in \mathcal{C},
\end{aligned} \tag{3.18}$$

where γ_1, γ_2, and γ_3 are positive weighting factors. In addition, the desired force profile $f^{\text{des}}(t)$ represents the optimal GRFs (i.e., inputs u) prescribed by the high-level data-driven planner in (3.12). 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., $h^{\text{des}}(t)$ can be hand-tuned to produce stable locomotion as was the case in Chapter 2. However, we aim to provide an optimal trajectory produced by a trajectory planner to reduce the required expertise necessary to enable stable locomotion and to provide additional robustness when compared to a heuristic trajectory.

3.5 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.

3.5.1 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. 3.2. 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 3.4.2. From the low-level QP (3.18), 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 3.4.2, a different low-level controller can be used as long as the outputs can be properly estimated, either directly through the controller, using an estimator, or measuring the real forces through additional sensors. As mentioned in Section 3.3.2, 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 complex and 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 overall approximation. This is in line with the promising results of [172, 173] for the control of nonlinear systems. Namely, [172] considers drone dynamics that are similar to the SRB model, and [173] attempts to dampen oscillations in power systems.

Although the size of the problem is reduced considerably by using (3.9) as opposed to (3.5), it is still large with 550 decision variables and 800 constraints. The size of the problem is particularly exacerbated by the fact that the final model $\mathcal{G}$ is dense, which significantly slows down the solvers. This issue will be discussed further in Chapter 5. The planner is solved

using OSQP [190] 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. Therefore, we run the planner every 30 (ms) and use the first three steps of the optimal predicted COM trajectory and GRFs as inputs passed to the low-level controller. Finally, the parameters in the predictive controller are taken to be $Q = \text{diag}(8e6, 5e5, 5e5, 5e3, 8e6, 8e5, 8e5, 5e3, 5e3, 5e5)$ and $R = 0.5I$, where I is the identity of appropriate size.

Remark 3.7. If good force estimates are not available, the chosen I-O pair seems restrictive. To alleviate this, one could also consider using $u := \text{col}(\text{z}^{\text{des}}, \dot{\text{x}}^{\text{des}}, \dot{\text{y}}^{\text{des}}, \dot{\text{z}}^{\text{des}}, \alpha^{\text{des}}, \omega^{\text{des}})$, $y := \text{col}(\text{z}, \dot{\text{x}}, \dot{\text{y}}, \dot{\text{z}}, \alpha, \omega)$ as the I-O pair for (3.12), which is less restrictive in terms of readily available measurements.

3.5.2 Data-Driven Experimental Results

The purpose of this section is to provide the parameters for the QP-based low-level controller (3.18) 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 [187] and takes approximately 0.22 (ms) using the same external laptop as the planner. Snapshots of various experiments using the trajectory planner in tandem with the low-level controller can be found in Fig. 3.3. In these experiments, the robot is commanded to blindly walk forward at 0.5 (m/s) and was subject to pushes (Fig. 3.3(a)), pulls (Fig. 3.3(b)), unknown rough terrain (Fig. 3.3(c)), and unstructured outdoor environments (Fig. 3.3(d)). In all scenarios, the quadruped was able to robustly maneuver. Videos of the experiments can be found online at [10]. Phase portraits for these stable gaits can be found in Fig. 3.4. 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. 3.4, Fig. 3.5 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 to provide velocity commands. The comparison between the output of the planner and the commanded velocities for a trot gait can be found in Fig. 3.6(a), and for a walk gait in Fig. 3.6(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. The videos of these experiments can be found online at [10]. 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

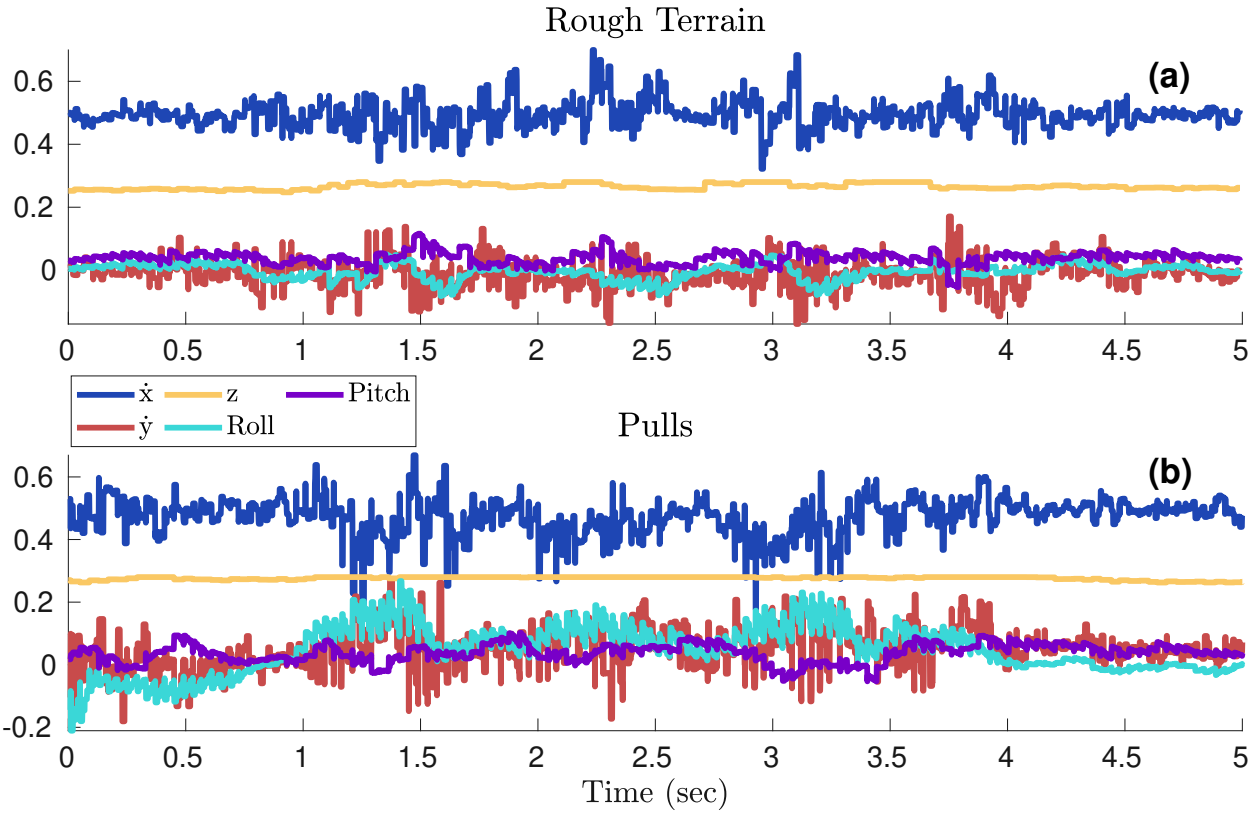

Figure 3.5: The prescribed trajectory from the planner while trotting subject to (a) rough terrain consisting of unstructured wooden blocks and (b) tethered pulls. The robot is commanded to walk forward at 0.5 (m/s), the height command is 0.28 (m), and all other states are commanded to be zero. Pulls occur for the first 4 seconds.

than that which was used during collection. However, dynamic gaits like bounding are likely to require additional data collection.

3.5.3 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. 3.7. The proposed approach, using only data to construct a model, performs comparably to a moderately tuned linearized SRB-based MPC. More specifically, the SRB-based MPC is linearized successively using the variational-based approach found in [90]. 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. However, we remark that the GRF profile for the

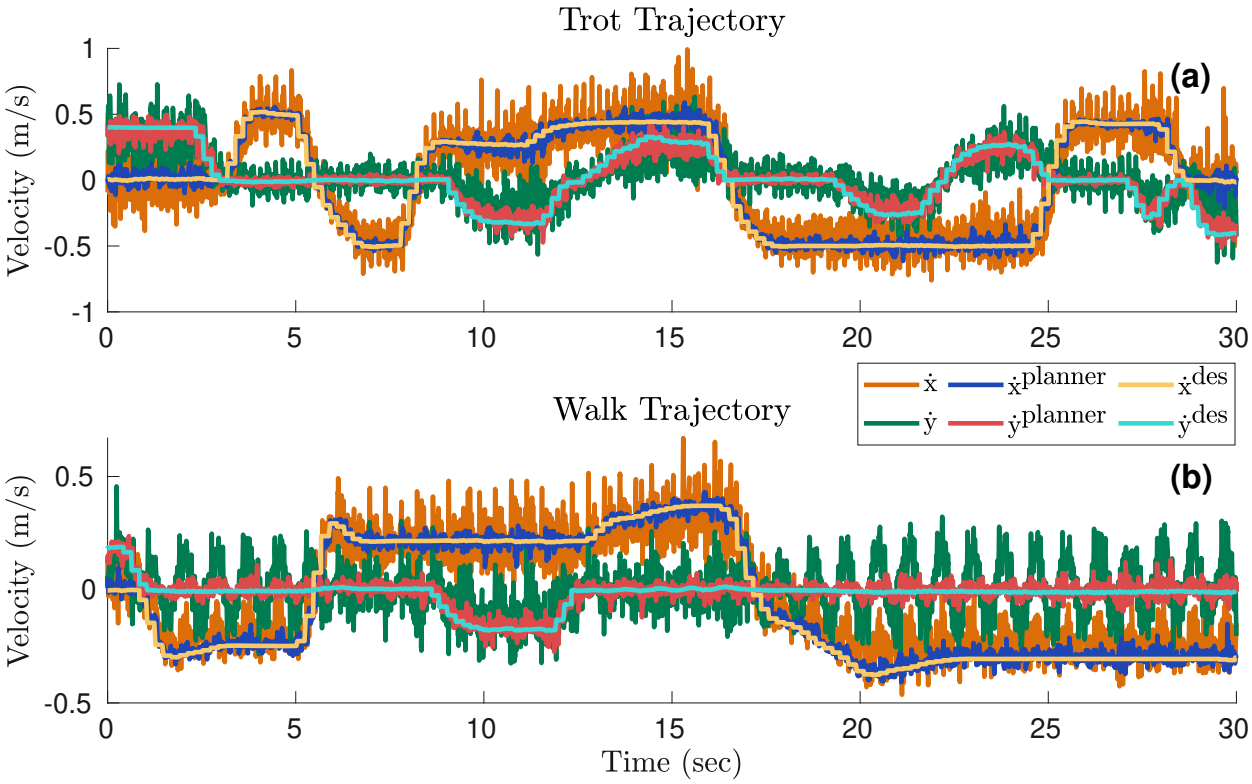

Figure 3.6: The figure shows the stable output tracking of the planner compared to the time-varying reference provided by a user through a joystick and the robot's actual states while using (a) a trot gait and (b) a walk gait. Each domain lasts 200 (ms).

data-driven approach tends to be more noisy and oscillatory when compared to the physics-based approach. Although 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 while also foregoing the need for explicit system identification. Improvements could potentially be obtained by considering a Page matrix representation [174] or singular value truncation [191], but we leave this to future investigation. Videos of the comparison can be found in [10].

3.5.4 Unknown Low-Level Controller

In this section, the same planner is applied to the unknown and stock (i.e., manufacturer's) controller of the A1 robot, with only very slight modifications for implementation. In particular, when using the A1 robot for only high-level control, we no longer have access to the scheduled foot contacts nor the estimated forces, both of which were previously obtained using the QP-based low-level controller in (3.18). However, we may still obtain contact information from the sensors at the feet. In order to use (3.13), we make the assumption

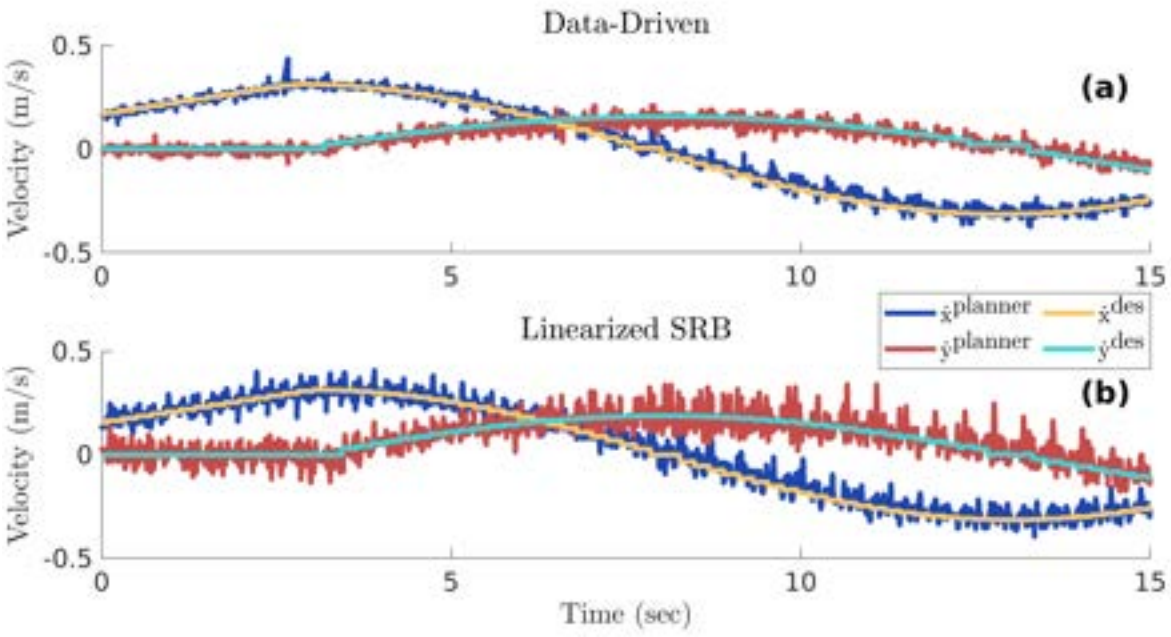

Figure 3.7: Hardware experiments showing the evolution of the trajectory produced by the data-driven planner (a) and a MPC planner using a linearized SRB model (b). The robot aims to follow a velocity profile that results in a circular path.

that the stock controller changes continuous-time domains at a fixed frequency, which is approximated based on the contact data collected during trotting. Furthermore, when considering the QP-based low-level controller, u_{ini} in (3.12) could be determined from the actual estimated forces used by the low-level controller. Without access to these estimates, u_{ini} is packed with the previous reduced-order forces prescribed by the trajectory planner, resulting in open-loop force planning. Even though the forces are open-loop when no force estimates are available, the reduced-order forces are comparable to the closed-loop case, as shown in Fig. 3.8. Based on Fig. 3.9, it can further be observed that the resulting trajectory remains near the desired commands. The planner is stable when used in either open- or closed-loop; this implies that the planner is agnostic to the low-level controller, assuming that the given low-level controller can track the provided trajectory sufficiently well. However, as in most scenarios, the closed-loop system is preferred since closed-loop systems generally provide a higher degree of robustness to disturbances. Furthermore, it is evident from Fig. 3.8 and 3.9 that the open-loop implementation is noisier and oscillates with a higher amplitude.

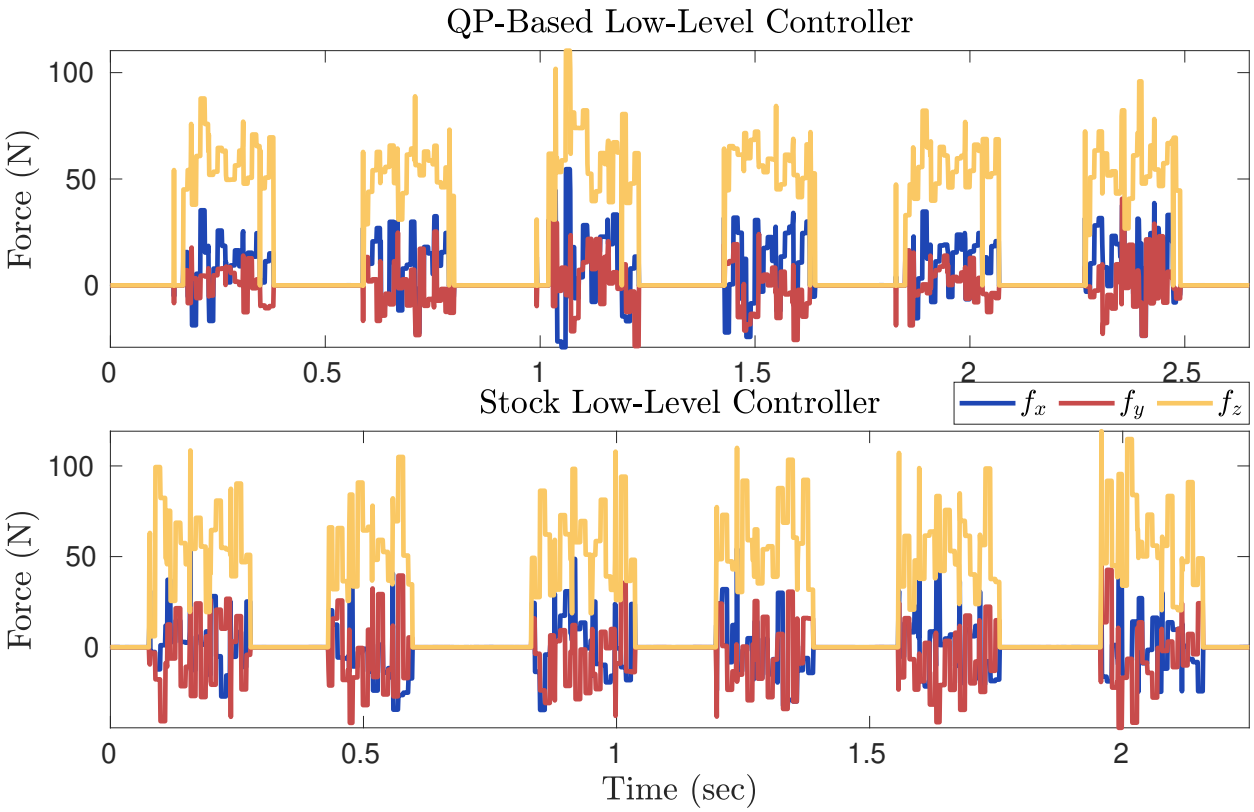

Figure 3.8: The reduced-order GRFs produced by the path-planner for the front right leg. The forces are tracked by the low-level controller in the case of the known controller (top), and the forces are neglected when using the unknown controller. The desired forward velocity is 0.5 (m/s).

3.6 Summary

This chapter 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

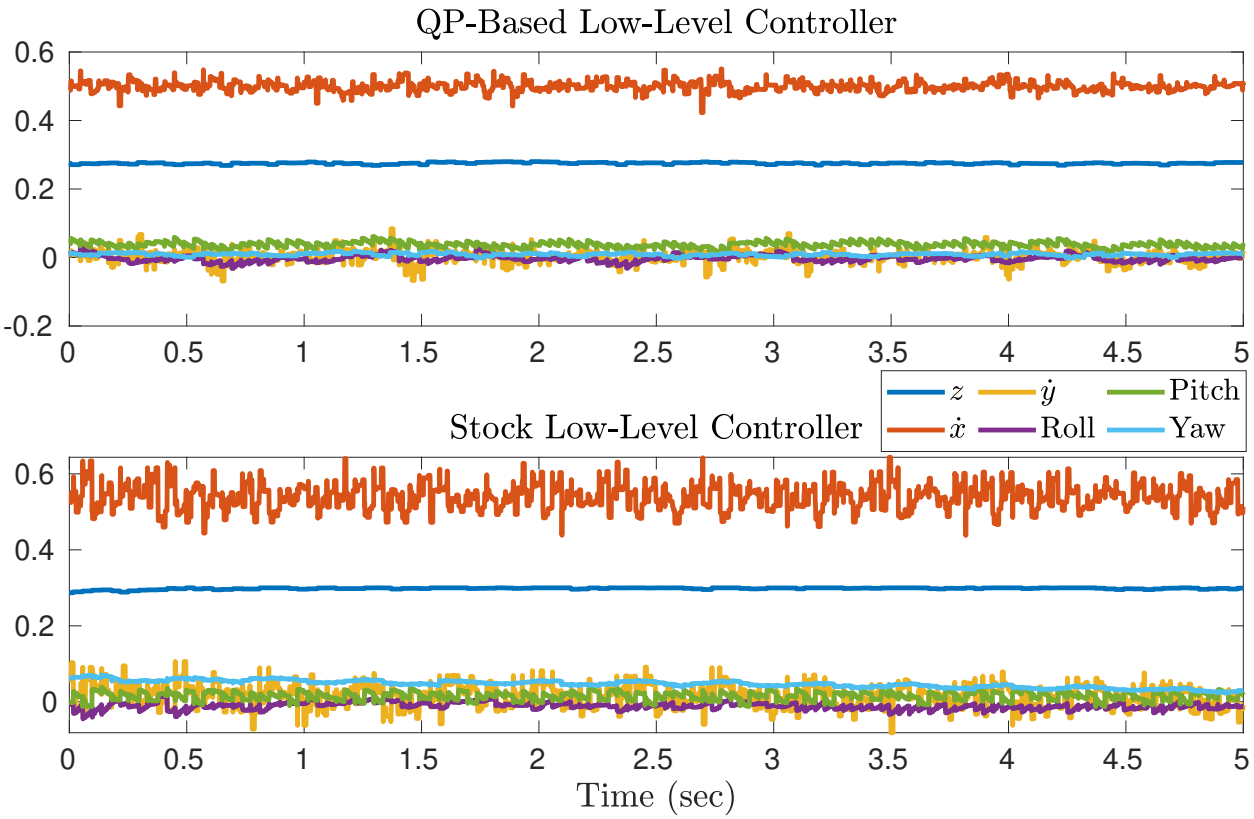

Figure 3.9: The prescribed trajectory from the high-level planner during steady-state when using a known low-level controller (top) and using an unknown low-level controller (bottom). When using the unknown low-level controller, the MPC runs in open-loop. The desired forward velocity is 0.5 (m/s), the desired height is 0.28 (m), and the desired value for all other states is 0. The height is in meters, $\dot{x}$ and $\dot{y}$ are in (m/s), and roll, pitch, and yaw are in radians. All states remain near their commanded values.

proposed method.

Chapter 4

Data-Driven Predictive Control for Cooperative Locomotion

4.1 Introduction

This chapter investigates multi-agent systems that are composed of high-dimensional subsystems, namely quadrupedal robots. These multi-agent teams are rigidly holonomically constrained to one another through the use of links connected via ball joints (see Fig. 4.1). During nominal locomotion, wherein the robots are traversing flat ground, the interaction forces remain relatively low. This is particularly the case in the event that the ground contacts for each robot are in sync, which is often approximately the case when considering time-based contact switching (i.e., switching at a common fixed frequency). However, when operating over daunting terrain or when subject to unknown external disturbances, the team of robots has the potential to exhibit high interaction wrenches that can lead to instability.

Current state-of-the-art approaches, even when considering only a single agent, generally involve using a reduced-order model of some kind [95]. 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 or trajectory planning. The goal of this chapter, therefore, is to address these issues. Namely, we aim to synthesize data-driven template models for trajectory planning for large multi-agent systems such that the resulting planner is computationally efficient for use in real time. In doing so, we also aim to obtain a reduced-order model with sufficient information about the interconnected system such that it can properly create trajectories for each agent. Furthermore, to the best of the authors' knowledge, there have not been implementations for multi-agent systems using the data-driven methods presented here, namely, behavioral systems theory. This work is an extension of [177] to *distributed* planning for constrained *multi-agent* teams of quadrupeds, which introduces many difficulties, including high interaction forces, the hybrid nature of legged locomotion, and unilateral constraints.

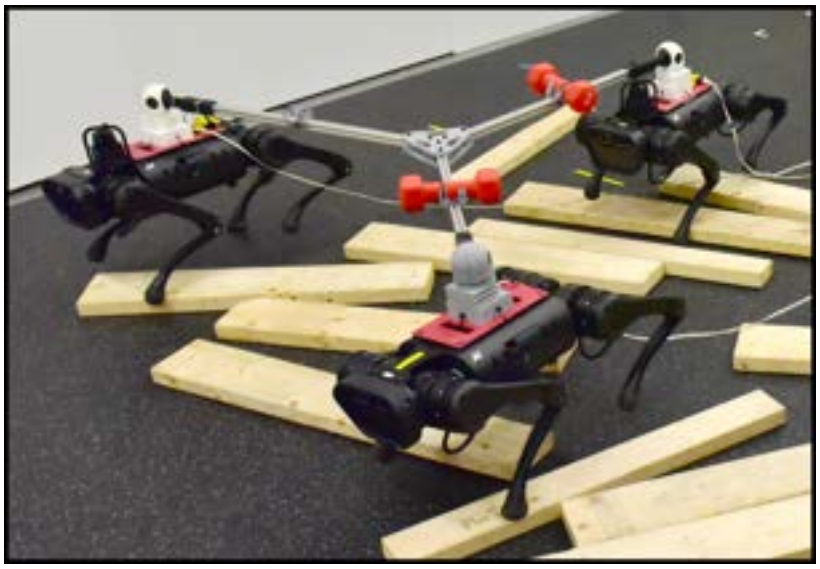

Figure 4.1: Snapshot showing the locomotion of three holonomically constrained A1 robots on wooden blocks.

The *overarching goal* of this chapter is to develop a computationally tractable real-time Data-Driven Predictive Controller (DDPC) for collaborative legged locomotion using a behavioral approach. Namely, this chapter'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. 4.2). 3) Simulation results for 5 constrained quadrupeds in the presence of ground height uncertainty and wavering terrain 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. The majority of the contents of this chapter are taken from our previous work [154].

4.2 Preliminaries

In this section, we recap some important concepts from behavioral systems theory that will be used throughout this chapter. 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

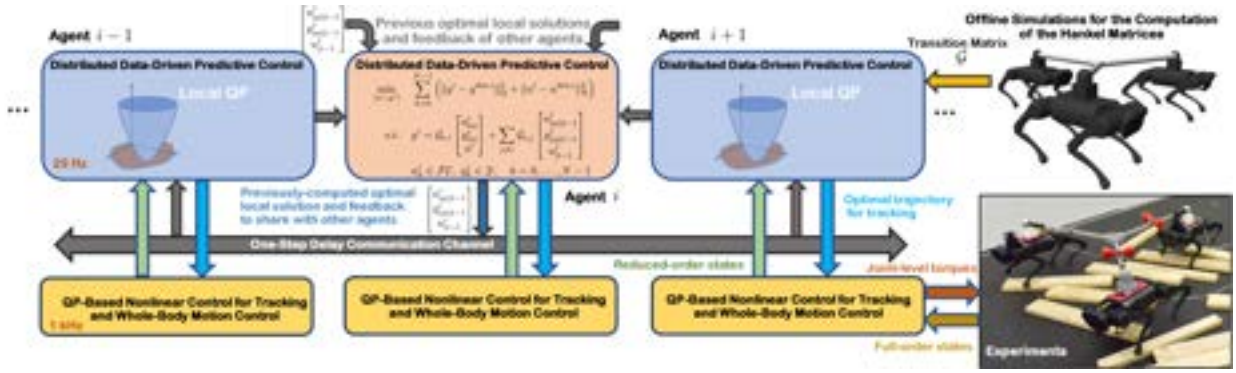

Figure 4.2: Overview of the proposed control algorithm with distributed DDPC algorithms at the high level for trajectory optimization of cooperative locomotion and nonlinear controllers at the low level for tracking and whole-body motion control.

time as follows:

$$\begin{aligned} x_{k+1} &= A\,x_k + B\,u_k \\ y_k &= C\,x_k + D\,u_k, \end{aligned} \tag{4.1}$$

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 (4.1) 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

$$\mathscr{H}_L(u^{\mathrm{d}}) := \begin{bmatrix} u_0^{\mathrm{d}} & u_1^{\mathrm{d}} & \cdots & u_{T-L}^{\mathrm{d}} \\ u_1^{\mathrm{d}} & u_2^{\mathrm{d}} & \cdots & u_{T-L+1}^{\mathrm{d}} \\ \vdots & \vdots & \ddots & \vdots \\ u_{L-1}^{\mathrm{d}} & u_L^{\mathrm{d}} & \cdots & u_{T-1}^{\mathrm{d}} \end{bmatrix} \in \mathbb{R}^{mL\times(T-L+1)}. \tag{4.2}$$

Definition 4.1 ([171])**.** 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 4.2 ([171])**.** The sequence $\{(u_k, y_k)\}_{k=0}^{T-1}$ is said to be a *trajectory* of the LTI system (4.1) 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 (4.1).

Using Definitions 4.1 and 4.2, we can now present a foundational theorem in behavioral systems theory, used to represent an LTI system based on observed trajectories.

Theorem 4.3. *[168, Theorem 1] Let an observed trajectory of (4.1), 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*

$$\begin{bmatrix} \mathscr{H}_L(u^d) \\ \mathscr{H}_L(y^d) \end{bmatrix} g = \begin{bmatrix} \bar{u} \\ \bar{y} \end{bmatrix}. \tag{4.3}$$

Theorem 4.3 provides a constructive manner in which an LTI system can be represented through its trajectories without explicit 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 (4.3). In contrast, the prediction horizon represents how far into the future predictions of the trajectories are made, similar to that of traditional MPC. Finally, we define $L = T_{\text{ini}} + N$ for compact notation. We denote the collected I-O data by $(u^{\text{d}}, y^{\text{d}})$ and can decompose the Hankel matrices of (4.3) into two parts as follows:

$$\mathscr{H}_L(u^{\text{d}}) = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \quad \mathscr{H}_L(y^{\text{d}}) = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}, \tag{4.4}$$

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 and sufficient condition to establish that the Hankel matrix is sufficiently rich is to choose T such that $T \geq (m+1)(T_{\text{ini}} + N + n) - 1$, which is a well-known result in behavioral systems theory.

4.3 Distributed DDPC for Trajectory Planning

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

4.3.1 Overview of DeePC

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 [172, 173] as follows:

$$\begin{aligned} \min_{(u,y,g,\sigma)} \quad & \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\text{des}}\|_Q^2 + \|u_k\|_R^2 \right) + \lambda_g \|g\|^2 + \lambda_\sigma \|\sigma\|^2 \\ \text{s.t.} \quad & \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g + \begin{bmatrix} 0 \\ \sigma \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix} \\ & u_k \in \mathcal{U}, \quad y_k \in \mathcal{Y}, \quad k = 0, \dots, N-1, \end{aligned} \tag{4.5}$$

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 the feasible input and output sets, respectively. In our notation, $(u_{\text{ini}}, y_{\text{ini}})$ 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 [172, 173]. 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 4.3 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 [177]. To circumvent the problem, we adopt an offline approximation for g, analogous the previous chapter, and to [173, 177] as follows:

$$g = \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}, \; y = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}, \; \mathcal{G} := Y_f \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger, \tag{4.6}$$

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 (4.5), considerably reducing the number of decision variables.

4.3.2 Distributed Multi-Agent Trajectory Planning

Here, we outline how the data-driven model (4.6) 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 (4.6) as follows:

$$\begin{aligned} \min_{(u,y)} \quad & \sum_{k=0}^{N-1} \left(\|y_k - y_k^{\text{des}}\|_Q^2 + \|u_k\|_R^2 \right) \\ \text{s.t.} \quad & y = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} \\ & u_k \in \mathcal{U}, \ y_k \in \mathcal{Y}, \qquad k = 0, \dots, N-1. \end{aligned} \tag{4.7}$$

This method has been shown to be amenable to trajectory planning for single-agent legged robots [177] 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. This effectively incorporates not only the dynamics reduced-order dynamics of individual agents but further includes information about the interconnections. In particular, Theorem 4.3 considers the trajectory $(u^{\text{d}}, y^{\text{d}})$ to define the Hankel matrix. To construct the Hankel matrix 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 in a centralized representation. Furthermore, this new Hankel matrix can be decomposed according to (4.4), and the corresponding g vector can be approximated using (4.6).

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 $\mathcal{G}$ into each agent's primary dynamics and the coupling dynamics between agents as follows:

$$\mathcal{G} = \begin{bmatrix} \mathcal{G}_{1,1} & \mathcal{G}_{1,2} & \cdots & \mathcal{G}_{1,n_a} \\ \mathcal{G}_{2,1} & \mathcal{G}_{2,2} & \cdots & \mathcal{G}_{2,n_a} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{G}_{n_a,1} & \mathcal{G}_{n_a,2} & \cdots & \mathcal{G}_{n_a,n_a} \end{bmatrix}, \tag{4.8}$$

where $\mathcal{G}_{i,j}$ represents the effect of agent j on the predicted output of agent i (i.e., coupling) and $\mathcal{G}_{i,i}$ represents the primary dynamics of agent i. In particular, the predicted output of

agent $i \in \mathcal{I}$ can be expressed by

$$y^i = \mathcal{G}_{i,i} \begin{bmatrix} u^i_{\text{ini}} \\ y^i_{\text{ini}} \\ u^i \end{bmatrix} + \sum_{j \neq i} \mathcal{G}_{i,j} \begin{bmatrix} u^j_{\text{ini}} \\ y^j_{\text{ini}} \\ u^j \end{bmatrix}, \tag{4.9}$$

where $(\cdot)^i$ represents the variables related to the ith agent. In essence, (4.9) is structured similarly to the control trajectory planner in the case of a single agent discussed in Chapter 3 but with the addition of a coupling term. However, (4.9) assumes knowledge of the *current* solution for all other agents, which cannot feasibly be obtained since each agent is running its own planner in a parallelized manner. In order 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

$$y^i \approx \mathcal{G}_{i,i} \begin{bmatrix} u^i_{\text{ini}} \\ y^i_{\text{ini}} \\ u^i \end{bmatrix} + \sum_{j \neq i} \mathcal{G}_{i,j} \begin{bmatrix} u^j_{\text{ini}|t-1} \\ y^j_{\text{ini}|t-1} \\ u^j_{|t-1} \end{bmatrix}, \tag{4.10}$$

where $u^j_{|t-1}$ denotes the optimal solution of the local DDPC for agent j at time $t-1$ and the summation is constant until the next time the DDPC is updated. In addition, $(u^j_{\text{ini}|t-1}, y^j_{\text{ini}|t-1})$ 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, cast into the form of local QPs to be solved at time t

$$\begin{aligned} \min_{(u^i, y^i)} \quad & \sum_{k=0}^{N-1} \left(\|y^i_k - y^{\text{des},i}_k\|^2_Q + \|u^i_k - u^{\text{des},i}_k\|^2_R \right) \\ \text{s.t.} \quad & y^i = \mathcal{G}_{i,i} \begin{bmatrix} u^i_{\text{ini}} \\ y^i_{\text{ini}} \\ u^i \end{bmatrix} + \sum_{j \neq i} \mathcal{G}_{i,j} \begin{bmatrix} u^j_{\text{ini}|t-1} \\ y^j_{\text{ini}|t-1} \\ u^j_{|t-1} \end{bmatrix} \\ & u^i_k \in \mathcal{U}, \; y^i_k \in \mathcal{Y}, \quad k = 0, \dots, N-1, \end{aligned} \tag{4.11}$$

where $u^{\text{des},i}_k$ and $y^{\text{des},i}_k$ represent the desired inputs and outputs for agent i at the prediction step k (see Fig. 4.2). 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 GRFs as the control inputs u. This will be clarified more in Section 4.5.1. 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.

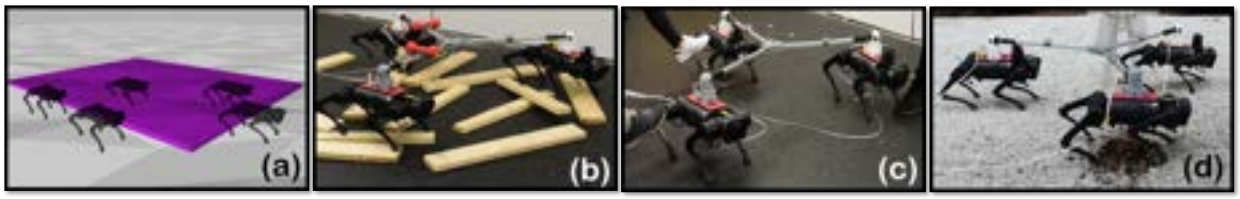

Figure 4.3: (a) Simulation results of 5 agents over varying terrain with a payload of 10 (kg), (b) rough terrain experiment with unstructured wooden blocks and a 4.5 (kg) payload, (c) experiment with push disturbances, and (d) experiment maneuvering over gravel. Videos are available online [11].

Remark 4.4. It is assumed The interaction wrenches between the agents are encapsulated in the data-driven model of the distributed planner. To simplify the synthesis of the low-level controllers, we do not consider the interaction wrenches between the agents when performing whole-body control.

4.4 Nonlinear Low-Level Controller

The purpose of this section is to briefly provide the details of the low-level controller [43], based on QP and virtual constraints [40], 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 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

$$M(q)\,\ddot{q} + H(q,\dot{q}) = \Upsilon\,\tau + J^{\top}(q)\,f, \tag{4.12}$$

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 [48]. Here, $h_0(q)$ denotes the controlled variables consisting of the COM position, orientation, and the Cartesian coordinates of the swing leg 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 [189, 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 [43]

$$\begin{aligned} \min_{(\tau,f,\delta)} \quad & \frac{\gamma_1}{2}\|\tau\|^2 + \frac{\gamma_2}{2}\|f - f^{\text{des}}\|^2 + \frac{\gamma_3}{2}\|\delta\|^2 \\ \text{s.t.} \quad & \ddot{h}(\tau, f) = -K_P\, h - K_D\, \dot{h} + \delta \quad \text{(Output Dynamics)} \\ & \ddot{r}(\tau, f) = 0 \qquad\qquad\qquad\quad \text{(No slippage)} \\ & \tau \in \mathcal{T}, \quad f \in \mathcal{FC} \qquad\quad \text{(Feasibility)}, \end{aligned} \tag{4.13}$$

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 (4.11). 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 [103, 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.

4.5 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. 4.3 and [127]). In simulations, this is imposed via a distance constraint.

4.5.1 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 of a holonomically constrained team of robots. This

is in contrast to [177], which used hardware data obtained by using a single agent. 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. Furthermore, explicitly deriving the interaction forces between agents is not scalable as the physics-based representation becomes increasingly nonlinear and complex as the cooperative team grows in size. Finally, the use of simulation data also displays 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 $y^{\mathrm{d}} = \mathrm{col}(\mathrm{z}, \dot{\mathrm{x}}, \dot{\mathrm{y}}, \mathrm{roll}, \mathrm{pitch}, \omega_z)$, where z is the standing height, $\dot{\mathrm{x}}$ and $\dot{\mathrm{y}}$ 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 team of quadrupeds are commanded to walk around in RaiSim [188] using just the low-level controller (4.13), 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

$$u_{k,\ell}^{\mathrm{des}} := \begin{cases} \mathrm{col}\left(0, 0, \dfrac{m^{\mathrm{net}} g_0}{N_{c,k}}\right), & \ell \in \mathcal{C}_k \\ \mathrm{col}(0,0,0), & \text{Otherwise,} \end{cases} \tag{4.14}$$

for each contacting leg $\ell \in \mathcal{C}_k^i$ and zero otherwise, where $m^{\mathrm{net},i}$ is the total mass of agent i, g_0 is the gravitational acceleration, $N_{c,k}^i$ is the anticipated number of contacting legs for agent i at time k, and $\mathcal{C}_k^i$ 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 $T_{\mathrm{ini}} = 10$ 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 (4.5) could pose an issue due to an increase in decision variables. However, this is mitigated by utilizing the approximation (4.6). In addition, for larger numbers of agents, it has also been observed that the model contains over-fitting and benefits greatly from singular value truncation [191] such that the model only encapsulates the dominant traits of the dynamics. For smaller numbers of agents, this does not pose an issue, but for 4 or more agents, we obtain better results when truncating

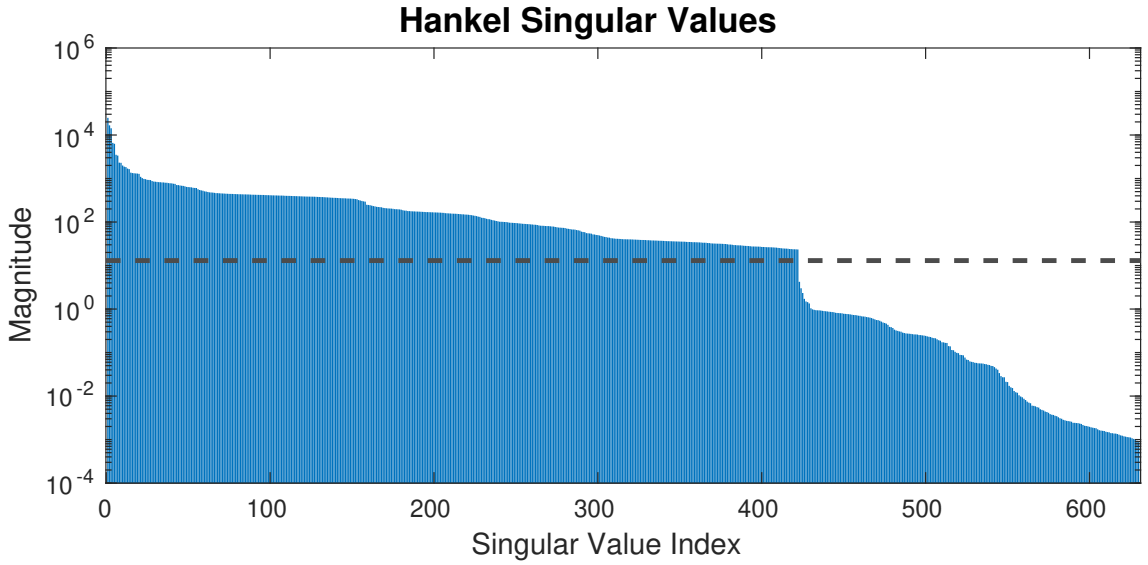

Figure 4.4: Example of the singular values of the Hankel matrix creates for a graph of 5 agents.

the resulting model using a Singular Value Decomposition (SVD) approach. The threshold is chosen differently based on the number of agents and the data collected, but in all scenarios, the threshold is apparent when plotting the singular values on a logarithm scale. In particular, the threshold is taken as the point at which the logarithm of the singular values exhibits a dramatic drop-off (see e.g., Fig. 4.4). However, for a far more in-depth discussion on choosing a cut-off such that the resulting system is a minimum realization, we refer the reader to [192].

4.5.2 Simulations

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. In addition, similar to the single agent case, the data-driven model is also dense, which further slows the solver down. Utilizing the distributed scheme, the predictive controller is updated every 40 (ms) (25 (Hz)), and the first 4 time steps are implemented, i.e., the prediction occurs over a horizon of 250 (ms), and the first 40 (ms) of the prediction is implemented. The distributed high-level planner is solved using OSQP [190] 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 ~ 30 (ms) have been observed in some configurations, further motivating the decision to update the planner at 40 (ms), which is conservative but still works well in practice. Conversely, the centralized approach cannot feasibly be solved in real time. The predictive controller parameters are

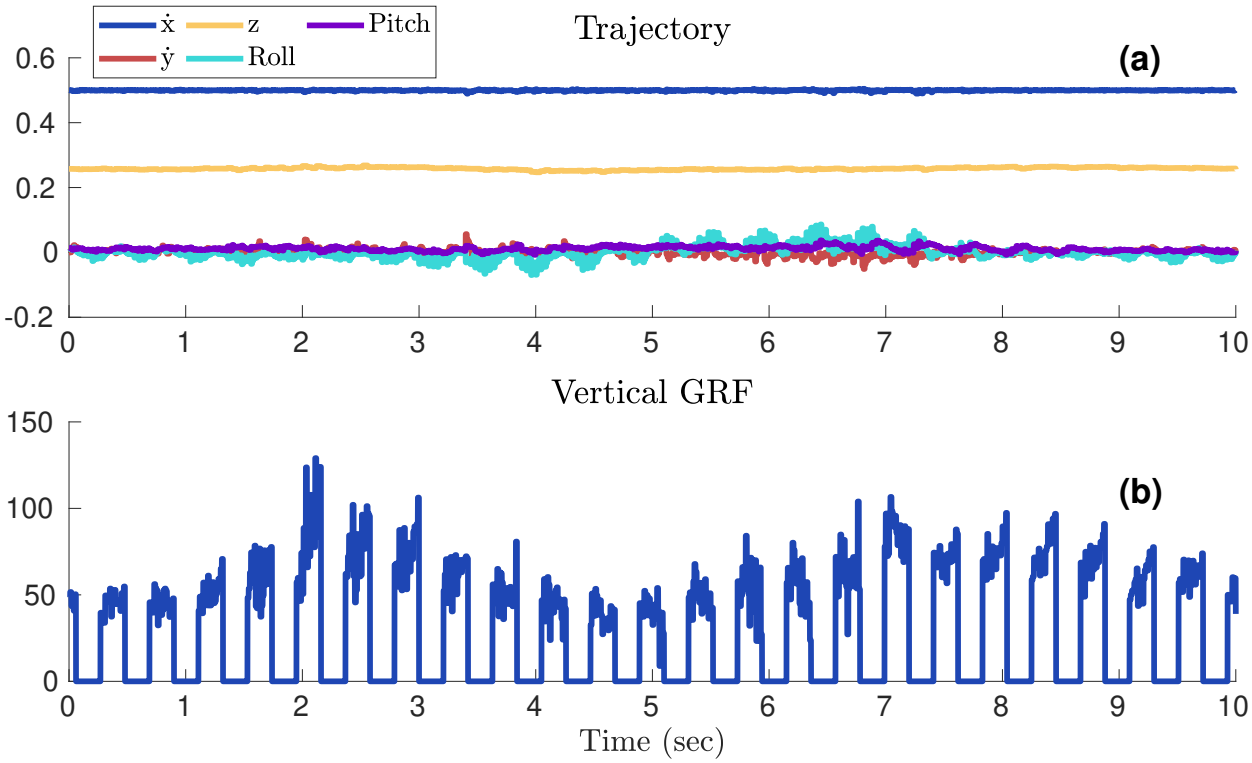

Figure 4.5: The trajectory from the planner (a) and the vertical GRF of the front right leg from the planner (b) for agent 1. A forward speed of 0.5 (m/s) is commanded, and the standing height is 0.26 (m). The multi-agent system is subject to uneven terrain and a payload of 10 (kg), and can maneuver robustly. A snapshot of the simulation can be found in Fig. 4.3 (a).

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 an identity matrix of an 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 for both the high- and low-level controllers are identical to those used during the hardware experiments in the following section.

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. 4.3 (a), while the prescribed forces and trajectory for the first agent can be found in Fig. 4.5. 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.

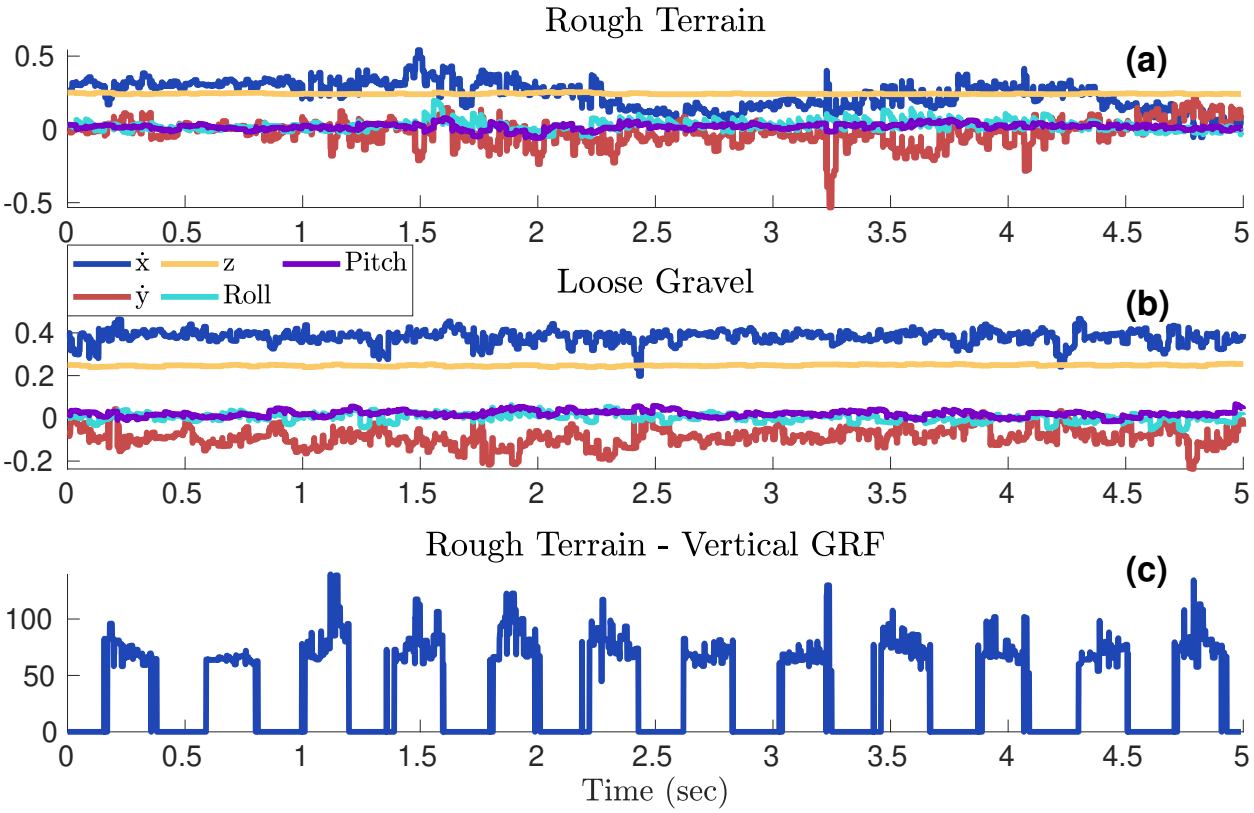

Figure 4.6: The trajectory from the planner of agent 1 while trotting at approximately 0.4 (m/s) subject to (a) rough terrain with unstructured wooden blocks and (b) loose gravel. We further show the vertical GRF for the front right leg produced by the planner for rough terrain in (c). The GRF during the gravel experiment is similar.

4.5.3 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. 4.3 (b)-(d), which shows the multi-agent system subject to external disturbances and unknown environments. In Fig. 4.6, we illustrate the behavior of the planner in terms of its prescribed trajectory when walking over unstructured wooden blocks (Fig. 4.6 (a)) and navigating over loose gravel (Fig. 4.6 (b)). The corresponding forces for the rough terrain experiment can be found in Fig. 4.6 (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. 4.7. 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 [11].

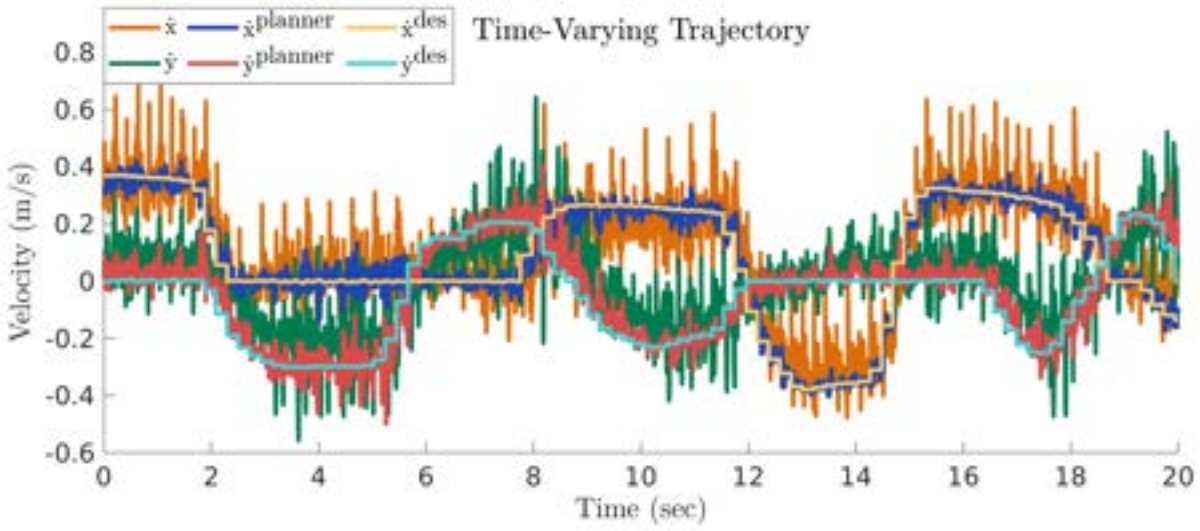

Figure 4.7: The plot shows the tracking performance of the planner when using a time-varying trajectory provided using a joystick. In this experiment, the quadrupeds navigate flat ground subject to a 6.8 (kg) payload. The plot shows the trajectory of agent 1.

4.6 Summary

This chapter 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.

Part III

Integration of Output-Space and State-Space for Trajectory Planning

Chapter 5

Combining Models for Multi-Agent Trajectory Planning

5.1 Introduction

The purpose of this chapter is to examine a different approach to utilizing behavioral systems theory for holonomically constrained multi-agent legged locomotion. In particular, it has been discussed in previous chapters that a major pitfall to the behavioral approach is the computational demand. This was, in part, remedied both through the use of a 2-norm approximation of the model as well as the use of a one-step communication delay to distribute computation in the multi-agent context. However, the fact that the data-driven problem formulation is dense still poses a considerable computational bottleneck. This is particularly the case since modern QP solvers can leverage the sparsity of a problem leading to far superior performance when compared to using a dense formulation. Therefore, the purpose of this chapter is largely similar to Chapter 4—creating a tractable and effective reduced-order model for holonomically constrained legged locomotion—with the additional goal of further reducing the computational burden by increasing sparsity. This chapter further aims to create a smoother gait and a better GRF profile compared to previous methods. In particular, we consider combining physics-based methods with behavioral systems theory, which, to the best of the authors' knowledge, is the first time that this has been done. The goal of Chapter 3 was to emulate the SRB model in hopes that a certain degree of the nonlinear dynamics could be captured in a neighborhood of the data collected by using a data-driven model. While this proved to be effective, the standard SRB model has consistently provided good results in a variety of scenarios. Furthermore, the data-driven approach led to a dense formulation that ran very slowly relative to a linearized SRB model. For these reasons, this last chapter aims to pivot and include the SRB dynamics directly into the formulation.

Although the SRB model has abstraction by not considering the legs or time-varying inertia and is successively linearized when used within a MPC framework, it is effective nonetheless. Consequently, we adopt a similar concept as Chapter 4 and provide an additional extension. Namely, when considering constrained multi-agent systems, we discussed that the nonlinear dynamics of each agent could be obtained with a sufficient degree of accuracy. Conversely, a reduced-order model that considers both the dynamics of individual agents and the interactions between them could not be obtained to an adequate degree of accuracy while also

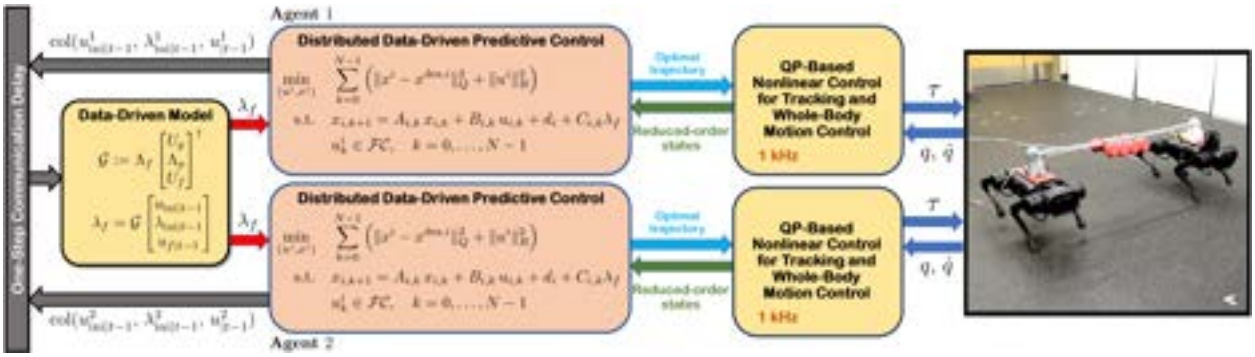

Figure 5.1: Block diagram of the SRB model used in tandem with a data-driven model that approximates the interaction forces between agents.

remaining feasible as a scalable reduced-order model. This is primarily due to the use of physics-based techniques. However, the reduced-order dynamics of individual agents can be represented very well through the SRB model. It then stands to reason that instead of parameterizing the entire reduced-order model through the use of a sizeable data-driven model as was done in Chapter 4, we can instead consider an interconnected SRB model where only the interaction wrench is defined using a data-driven model (see e.g., Fig. 5.1).

The proposed method is in contrast to previous work that aimed to parameterize the difference between reduced- and full-order models through the use of reinforcement learning [61]. This is further different from [103, 127] in that the interactions are modeled using data in the work presented here. Furthermore, this work considers a different linearization technique for the SRB model compared to both [127] and [61] that is simpler and leads to more robust locomotion, although it cannot address acrobatic motions. Additionally, the formulation provided in this chapter considers a very sparse linear predictive controller, leading to very fast solve times, whereas other works such as [128] have considered NMPC approaches to similar problems that are far less efficient in terms of computation time. Finally, [128] considers an extremely simple model, namely the unicycle model, which reduces the complexity of the system immensely and is not sufficiently accurate for robots such as quadrupeds.

5.2 Standard Single Rigid Body Model

This section provides an overview of the dynamics of the SRB model and the successive linearization technique used in tandem with the MPC formulation. In particular, the nonlinear

SRB dynamics in world coordinates for a single agent are defined by [87]

$$\begin{aligned} \ddot{p}^{\text{com}} &= \frac{1}{m_{\text{tot}}} \sum_{i=1}^{4} f_i - \bar{g} \\ \dot{\omega} &= I^{-1} \left(\sum_{i=1}^{4} \hat{r}_i f_i - \hat{\omega} I \omega \right) \\ \dot{R} &= \hat{\omega} R, \end{aligned} \tag{5.1}$$

where $p^{\text{com}} \in \mathbb{R}^3$ represents the position of the center of mass, $f_i \in \mathbb{R}^3$ is the GRF of foot i, $r_i \in \mathbb{R}^3$ is the relative position of foot i with respect to the COM, $R \in \mathbb{R}^{3\times3}$ is the rotation matrix for the torso of the body, and $\omega \in \mathbb{R}^3$ is the angular velocity. Furthermore, m_{tot}, $I \in \mathbb{R}^{3\times3}$, and $\bar{g} \in \mathbb{R}^3$ represent the total mass, the inertia, and the gravity vector, respectively. Finally, in this notation, $\hat{(\cdot)} \in \mathbb{R}^{3\times3}$ represents the skew-symmetric operator. These dynamics are nonlinear and contain a rotation matrix as a state, making this formulation infeasible for use with a QP solver. For this reason, we opt to consider a linearized model that uses Euler angles. In particular, we adopt a yaw-pitch-roll convention for the Euler angles and consider the small-angle approximation for both the pitch and roll states, and further assume that the precession and nutation of the torso are negligible. These are reasonable assumptions for nominal locomotion, i.e., for relatively small body angles and angular velocities. For more information regarding these simplifications, we refer the interested reader to [87, 193]. With this approximation, we can then use the rate of change of Euler angles defined by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \omega, \tag{5.2}$$

where ϕ, θ, and ψ represent the roll, pitch, and yaw angles, respectively, and we define $\alpha = \text{col}(\phi, \theta, \psi)$ for future use. This expression then simplifies to $\dot{\alpha} = R_z^\top \omega$. Using this approximation, the linearized model is then given by

$$x_{k+1} = A_k x_k + B_k u_k + d \tag{5.3}$$

where $u_k \in \mathbb{R}^{12}$ is a stacked vector of the GRFs for all four legs. The state vector is taken to be $x = \text{col}(p^{\text{com}}, \dot{p}^{\text{com}}, \alpha, \omega)$, resulting in the following state matrices [87]:

$$A = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & R_z^\top \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \in \mathbb{R}^{12\times12}, \quad B = \begin{bmatrix} \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ \frac{1}{m_{\text{tot}}}\mathbf{I}_3 & \cdots & \frac{1}{m_{\text{tot}}}\mathbf{I}_3 \\ \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ I^{-1}\hat{r}_1 & \cdots & I^{-1}\hat{r}_4 \end{bmatrix} \in \mathbb{R}^{12\times12}, \quad d = \begin{bmatrix} 0_3 \\ 0_3 \\ \bar{g} \\ 0_3 \end{bmatrix} \in \mathbb{R}^{12},$$

where $\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ is an identity matrix, $\mathbf{0}_3 \in \mathbb{R}^{3\times3}$ is a matrix of zeros, and $0_3 \in \mathbb{R}^3$ is a vector of zeros. When used in the context of MPC, these matrices are often held constant

throughout the prediction horizon and are updated every time that the MPC is solved. However, some implementations make B varying throughout the horizon based on the desired trajectory of the COM since the relative distance between the foot and the COM can change considerably for large temporal horizons. This is not a strict requirement but can aid in obtaining more robust locomotion, particularly for dynamic gaits. In this work, all of the matrices are held constant throughout the prediction horizon and are sequentially updated each time the MPC runs.

5.3 Interconnected Model and Interaction Wrenches

In this section, we will present an interconnected SRB model that considers interaction forces between agents (see e.g., Fig 5.2). In addition, we discuss how the interaction forces can be obtained in closed form, which will be important when creating the data-driven model for two agents. We first define the connection point on agent i to be $p_i = p_i^{\text{com}} + R_i \bar{d}_i$, where $\bar{d}_i$ is the relative distance between the COM and the connection point and is constant for all agents. Furthermore, the velocity and acceleration of the interaction point are given by

$$\begin{aligned} \dot{p}_i &= \dot{p}_i^{\text{com}} + \hat{\omega}_i R_i \bar{d}_i \\ \ddot{p}_i &= \ddot{p}_i^{\text{com}} + \left(\hat{\dot{\omega}}_i R_i + \hat{\omega}_i \dot{R}_i \right) \bar{d}_i. \end{aligned} \tag{5.4}$$

We may then define the interconnected dynamics for the motion of each individual agent subject to interaction forces. The equations are presented for two agents but generalize to any number of agents with some minor additional algebraic manipulation. In particular, the two agent interconnected dynamics are given by

$$\begin{aligned} \ddot{p}_i^{\text{com}} &= \frac{1}{m_{\text{tot}}} \left(\sum_{k=1}^{4} f_{i,k} + \frac{p_i - p_j}{\|p_i - p_j\|} \lambda_{ij} \right) - \bar{g} \\ \dot{\omega}_i &= I^{-1} \left(\sum_{k=1}^{4} \hat{r}_{i,k} f_{i,k} + \widehat{R_i \bar{d}_i} \frac{p_i - p_j}{\|p_i - p_j\|} \lambda_{ij} - \hat{\omega}_i I \omega_i \right) \\ \dot{R}_i &= \hat{\omega}_i R_i, \end{aligned} \tag{5.5}$$

where $\lambda_{ij} \in \mathbb{R}$ represents the magnitude of the interaction force between agent i and j, and $(p_i - p_j)/\|p_i - p_j\|$ represents the normalized vector pointing from the connection point on agent i to the connection point on agent j. It is important to note that $\lambda_{ij} = \lambda_{ji}$ since this is only a magnitude of the interaction force and considers no direction information. By combining (5.4) and (5.5), one can obtain the nonlinear dynamics of the connection point. We can now define the rigid distance constraint between agents i and j as

$$\Delta_{ij} = (p_i - p_j)^\top (p_i - p_j) = \Delta_{ij|0}, \tag{5.6}$$

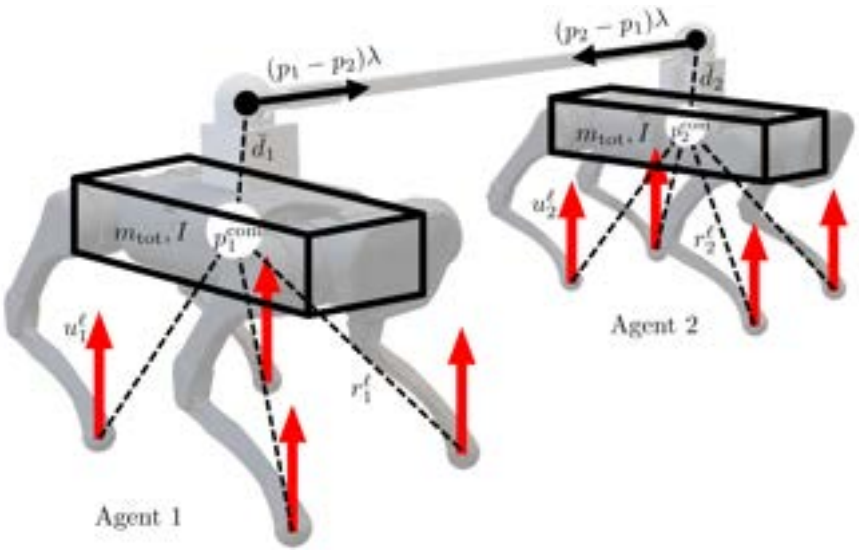

Figure 5.2: Visual representation of two SRB models subject to interaction forces. This concept extends to more agents as well.

where $\Delta_{ij|0}$ is the distance between the connection points and is constant for the duration of locomotion. Differentiating this holonomic constraint twice, the dynamics of the holonomic constraint are defined as

$$\begin{aligned}\dot{\Delta}_{ij} &= (\dot{p}_i - \dot{p}_j)^\top (p_i - p_j) = 0 \\ \ddot{\Delta}_{ij} &= (\ddot{p}_i - \ddot{p}_j)^\top (p_i - p_j) + (\dot{p}_i - \dot{p}_j)^\top (\dot{p}_i - \dot{p}_j) = 0.\end{aligned} \tag{5.7}$$

Using (5.7) together with (5.5) and (5.4), we are able to solve explicitly for the interaction forces λ_{ij} between agent i and j given their respective state vectors. Although we may obtain the interaction forces in closed form fairly simply for two agents, extending this to many agents is very complex and would require considerable computation time, motivating the decision to obtain the interaction forces via a data-driven model. However, this closed-form expression is used during the data collection for this control scheme due to limitations in the simulation environment considered. This will be discussed further in Section 5.5.

5.4 Integration of Physics- and Data-Based Models

With the interconnected SRB dynamics in hand (5.5), this section presents the linearized SRB dynamics subject to external forces in a centralized manner for two agents. In particular, this is an extension of (5.3) to the multi-agent setting while considering the interaction forces

and is given by

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}\tilde{u}_k + \tilde{C}\lambda_k + \tilde{d}. \tag{5.8}$$

In this notation, we define $\tilde{x}_k := \text{col}(x_{1,k}, x_{2,k})$ and $\tilde{u}_k := \text{col}(u_{1,k}, u_{2,k})$, which denotes the collective state and input for both agents, respectively. Furthermore, $\lambda_k \in \mathbb{R}$ represents the magnitude of the interaction force. Note that this is a simpler notation than what was considered in (5.5) and is used in the two agent case since $\lambda_{ij} = \lambda_{ji}$. We then define our state matrices to be

$$\tilde{A} = \begin{bmatrix} A_1 & \mathbf{0}_{12} \\ \mathbf{0}_{12} & A_2 \end{bmatrix} \in \mathbb{R}^{24\times 24}, \quad \tilde{B} = \begin{bmatrix} B_1 & \mathbf{0}_{12} \\ \mathbf{0}_{12} & B_2 \end{bmatrix} \in \mathbb{R}^{24\times 24}, \quad \tilde{C} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \in \mathbb{R}^{12}, \quad \tilde{d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \in \mathbb{R}^{12},$$

where A_i, B_i, and d_i are the matrices and vectors corresponding to agent i, and $\mathbf{0}_{12} \in \mathbb{R}^{12\times 12}$ is a matrix of zeros. Finally, C_i is the input distribution matrix for the interaction force. More specifically, we have

$$C_i = \begin{bmatrix} 0_3 \\ \frac{1}{m_{\text{tot}}} \frac{p_i - p_j}{\|p_i - p_j\|} \\ 0_3 \\ I^{-1}\hat{\tilde{d}}_i \frac{p_i - p_j}{\|p_i - p_j\|} \end{bmatrix} \in \mathbb{R}^{12} \quad \forall i \neq j.$$

This matrix maps the magnitude of the interaction force to both forces and torques induced about the COM, while it also adds a direction to the magnitude, where we may assume that the interaction force is applied along the vector pointing from one connection point to the other. This is a reasonable assumption since the robots are connected via ball joints, and the frictional force preventing rotation is negligible. By considering only the magnitude of the interaction force, we are able to reduce the number of variables that must be approximated using a data-driven model, as will be discussed more thoroughly in what follows.

In order to use (5.8) in a predictive control framework, one would need to obtain the interaction forces throughout the prediction horizon. This is particularly difficult considering the interaction forces nonlinearly depend on the state and the input. In turn, one would need to consider a successively linearized version of the closed-form expression or an NMPC formulation would be required [127, 128]. In order to alleviate this, we consider a data-driven approach to model the interaction forces. Consider the behavioral model given by

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ y_f \end{bmatrix},$$

for some choice of inputs denoted by u and some outputs denoted by y. In this notation, $(\cdot)_{\text{ini}}$ denotes the previous input-output pairs, and $(\cdot)_f$ represents the future predicted input-

output pairs. This is the same form as that which was provided in the previous chapters. We can then once again use the 2-norm approximation for the vector g and end up with a model of the form

$$y_f = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \end{bmatrix}.$$

Unlike the previous chapters, here we are not modeling the motion of the system. I.e., we are not trying to control the position or velocity of the system by using this data-driven model. To this end, we choose the input, u, to be the GRF at the contacting legs and take the output, y, to be the magnitude of the interaction force (i.e., λ). Since the output is taken to only be λ, we will move forward with the following model notation for the sake of transparency:

$$\lambda_f = \mathcal{G} \begin{bmatrix} u_{\text{ini}} \\ \lambda_{\text{ini}} \\ u_f \end{bmatrix}. \tag{5.9}$$

It is assumed that the data collected to create this model is done so on an interconnected system of two agents. Therefore, we have $\lambda_f \in \mathbb{R}^N$, $\lambda_{\text{ini}} \in \mathbb{R}^{T_{\text{ini}}}$, $u_f \in \mathbb{R}^{24N}$, and $u_{\text{ini}} \in \mathbb{R}^{24T_{\text{ini}}}$, where N is the prediction horizon, and T_{ini} is the estimation horizon. Note that as the number of agents increases, the size of $\lambda_{(\cdot)}$ depends explicitly on the configuration of the connection graph, while the size of $u_{(\cdot)}$ grows linearly with the number of agents. Using this model, we now have a mapping between the GRF of each agent and the magnitude of the interaction forces between them. In this case, we chose the inputs to the model to be only the GRF from each agent. However, one could also consider using a portion of the state as an input, such as incorporating the velocity or orientation of each agent. In contrast, it has been observed that including additional inputs does not have a considerable impact on performance. Moreover, it is particularly difficult to obtain a sufficiently persistently exciting signal for the COM states, which can result in degraded performance and, in many cases, instability. Finally, including additional states as inputs to the data-driven model would decrease the sparsity of the problem, thereby slowing down solving times, which directly contradicts one of the goals of this chapter.

Moving forward, we will adopt the notation such that $u_{f,k}$ denotes the GRF for both agents at prediction step k for the data-driven model, which is identical to $\tilde{u}_k$ in (5.8). Similarly, $\lambda_{f,k} = \lambda_k$ will denote the interaction force at prediction step k, and we will denote $\mathcal{G}_k$ to be the k^{th} row of the N-step state transition matrix $\mathcal{G}$. We may also decompose the $\mathcal{G}$ matrix into its future and previous portions. More specifically, using the new notation, we have

$$\lambda_k = \mathcal{G}_k^{\text{ini}} \begin{bmatrix} \tilde{u}_{\text{ini}} \\ \lambda_{\text{ini}} \end{bmatrix} + \mathcal{G}_k^u \tilde{u}_k, \tag{5.10}$$

where $\mathcal{G}_k^u$ and $\mathcal{G}_k^{\text{ini}}$ contain the columns of $\mathcal{G}$ corresponding to the input and the previous data, respectively. With this in hand, we may now combine (5.10) with (5.8), resulting in a

centralized model for collaborative locomotion given by

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + (\tilde{B} + \tilde{C}\mathcal{G}_k^u)\tilde{u}_k + \tilde{C}\mathcal{G}_k^{\text{ini}} \begin{bmatrix} \tilde{u}_{\text{ini}} \\ \lambda_{\text{ini}} \end{bmatrix} + \tilde{d}. \tag{5.11}$$

Analogous to Chapter 4, we now aim to distribute this model for computational purposes. We begin by decomposing $\mathcal{G}$ into each agent's local and coupling dynamics given by

$$\mathcal{G} = \begin{bmatrix} \mathcal{G}_{1,1} & \mathcal{G}_{1,2} \\ \mathcal{G}_{2,1} & \mathcal{G}_{2,2} \end{bmatrix}. \tag{5.12}$$

From here, we once again adopt a one-step communication delay protocol and obtain our distributed model as follows:

$$x_{i,k+1} = \underbrace{A_i\, x_{i,k} + (B_i + C_i\mathcal{G}_{ii,k}^u)\, u_{i,k} + C_i\mathcal{G}_{ii,k}^{\text{ini}} \begin{bmatrix} u_{i,\text{ini}} \\ \lambda_{i,\text{ini}} \end{bmatrix} + d_i}_{\text{Local Dynamics}} + \underbrace{C_j \sum_{i \neq j} \mathcal{G}_{ij,k} \begin{bmatrix} u_{j,\text{ini}|t-1} \\ \lambda_{j,\text{ini}|t-1} \\ u_{j,k|t-1} \end{bmatrix}}_{\text{Coupling Dynamics}}. \tag{5.13}$$

Using this approach, the sparsity of the dynamics (i.e., the equality constraints in the QP) increases dramatically compared to the approach considered in Chapter 4. Moreover, under this control scheme, the size of the problem in terms of the number of decision variables is unchanged relative to single-agent SRB-based MPC. However, one additional approximation is made such that the sparsity remains unchanged relative to the single-agent case using the standard linearized SRB model. In particular, we adopt a one-step communication delay on *all* of the input-output pairs, therefore making λ constant each time the predictive controller is run. It has been observed that this further results in more robust locomotion, easier tuning, and better tracking of the desired trajectory. More specifically, we have

$$\lambda = \mathcal{G} \begin{bmatrix} u_{\text{ini}|t-1} \\ \lambda_{\text{ini}|t-1} \\ u_{|t-1} \end{bmatrix}. \tag{5.14}$$

Our final model, which will be used during both simulations and experiments, is then described by

$$x_{i,k+1} = A_{i,k}\, x_{i,k} + B_{i,k}\, u_{i,k} + C_{i,k}\mathcal{G}_k \begin{bmatrix} u_{\text{ini}|t-1} \\ \lambda_{\text{ini}|t-1} \\ u_{|t-1} \end{bmatrix} + d_i, \tag{5.15}$$

where it is evident that the two rightmost terms are constant each time the planner is solved. Finally, with the model in hand, we may now present the trajectory planner that will be

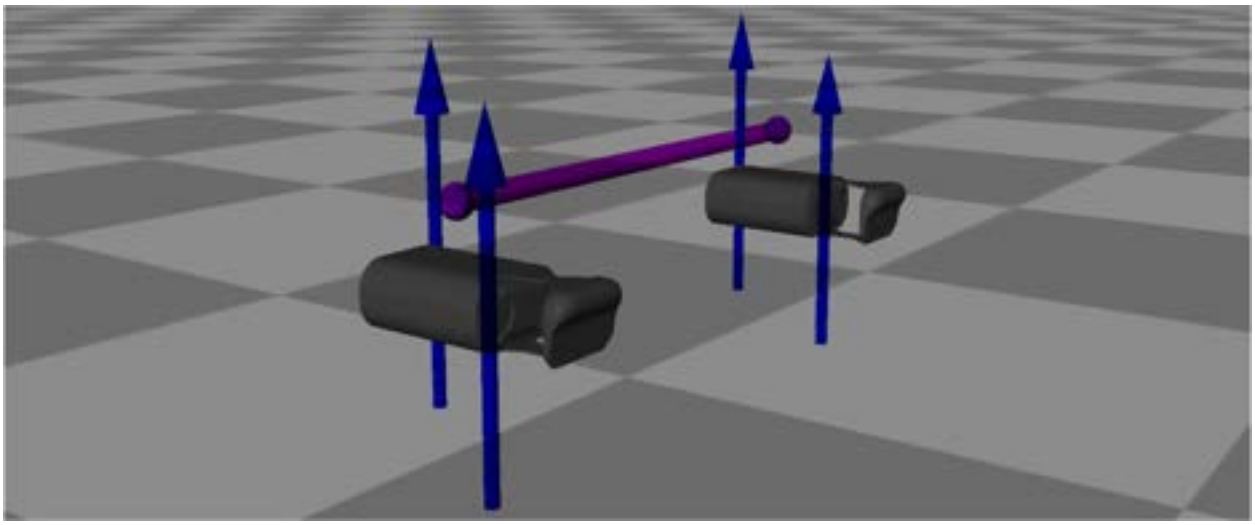

Figure 5.3: This figure displays the reduced-order simulation setup in RaiSim. The blue arrows represent the GRF, the purple spheres represent the ball joints at the connection point, and the purple cylinder represents the bar rigidly connecting the agents.

used during simulations and experiments

$$\begin{aligned} \min_{(x^i,u^i)} \quad & \sum_{k=0}^{N-1} \left(\|x_k^i - x_k^{\text{des},i}\|_Q^2 + \|u_k^i\|_R^2 \right) \\ \text{s.t.} \quad & x_{i,k+1} = A_{i,k}\, x_{i,k} + B_{i,k}\, u_{i,k} + C_{i,k} \mathcal{G} \begin{bmatrix} \tilde{u}_{\text{ini}|t-1} \\ \lambda_{\text{ini}|t-1} \\ \tilde{u}_{|t-1} \end{bmatrix} + d_i \\ & u_k^i \in \mathcal{U}, \quad k = 0, \ldots, N-1, \end{aligned} \tag{5.16}$$

where $x_k^{\text{des},i}$ denotes the desired trajectory for agent i at prediction step k. Furthermore, the feasible set $\mathcal{U}$ is taken to be 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.

5.5 Simulations and Experiments

The purpose of this section is to provide validation of the approach proposed in this chapter. This is done through a series of simulations and experiments, starting with only the reduced-order model, then proceeding with the full-order implementation.

5.5.1 Data Collection Procedure

Before moving to the experimental results, we must first provide the details of the data collection. Analogous to Chapter 4, the data is collected in the simulation environment RaiSim [188] for the interconnected system of robots. In contrast, the data collected here is based *only* on the reduced-order SRB model instead of the full-order model. In particular, two interconnected trunks are simulated without additional appendages, and external forces are applied on the COM of each respective trunk emanating from the desired foot location. See Fig. 5.3 for a representative example of the simulation setup. This effectively simulates *only* the interconnected SRB model. In order to maintain stability during the data collection, the model described in Section 5.2 is used in tandem with an MPC framework in a purely decentralized manner to calculate the forces required to propel the system forward. While this method does stabilize the network of robots, we aim to improve performance through the addition of a data-driven approximation of the interaction forces without adversely affecting the solve times. Finally, it should be noted that the data is collected while using a trot gait that has a step frequency of 5 Hz and is commanded to move forward with a desired velocity of 0.5 (m/s).

Since the input to the data-driven model is taken to be the GRF, a random perturbation of up to 10 (N) is added to the prescribed GRF at each time step during data collection in order to maintain persistence of excitation for the input. In terms of the output, RaiSim does not have a manner in which internal forces of a system can be obtained. Therefore, the closed-form expression for the interaction forces described in Section 5.3 is used to calculate the corresponding output of the data-driven model, λ, in an offline manner to construct the Hankel matrices. The lack of the ability to obtain the reaction forces from the simulator is a pertinent limitation of the study. In moving to a greater number of agents, it is imperative to have the ability to determine the interaction forces without requiring a closed-form expression, as the complexity of such equations increases dramatically with the number of agents. However, using a closed-form expression is not prohibitive for the two-agent case considered in this chapter.

5.5.2 Reduced-Order Simulations

This section discusses the initial validation of the proposed model in simulation using only the reduced-order model (see Fig. 5.3), and provides a comparison to using a strictly model-based approach. In particular, the GRF and trajectory for nominal locomotion can be found in Fig. 5.4 for agent 1, where it is evident that the system is able to maneuver over flat ground in a stable manner. In all of the simulations presented in this section, each robot is commanded to move forward at a velocity of 0.5 (m/s) with a height of 0.26 (m). All other states are commanded to remain zero. During nominal locomotion, the interaction forces remain small (less than 1 (N) on average) and do not substantially influence the gait. However, when external forces are applied to the system, the interaction forces increase

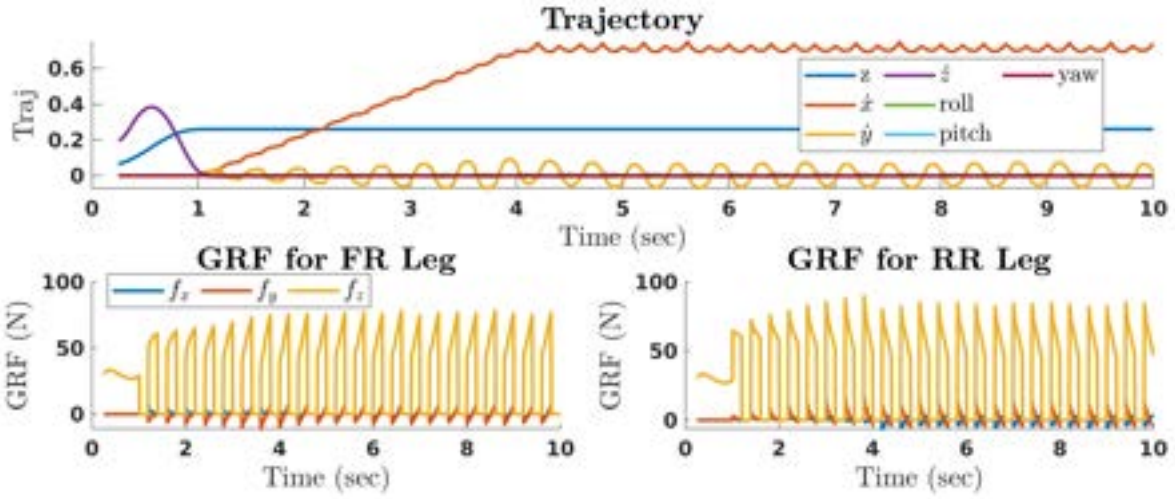

Figure 5.4: This figure shows the trajectory produced by the proposed MPC (top), the prescribed GRF for the front right leg (bottom left), and the prescribed GRF for the rear right leg (bottom left) for agent 1. The data pertaining to agent 2 follows a similar trend. Each agent aims to track a velocity of 0.75 (m/s) and a standing height of 0.26 (m).

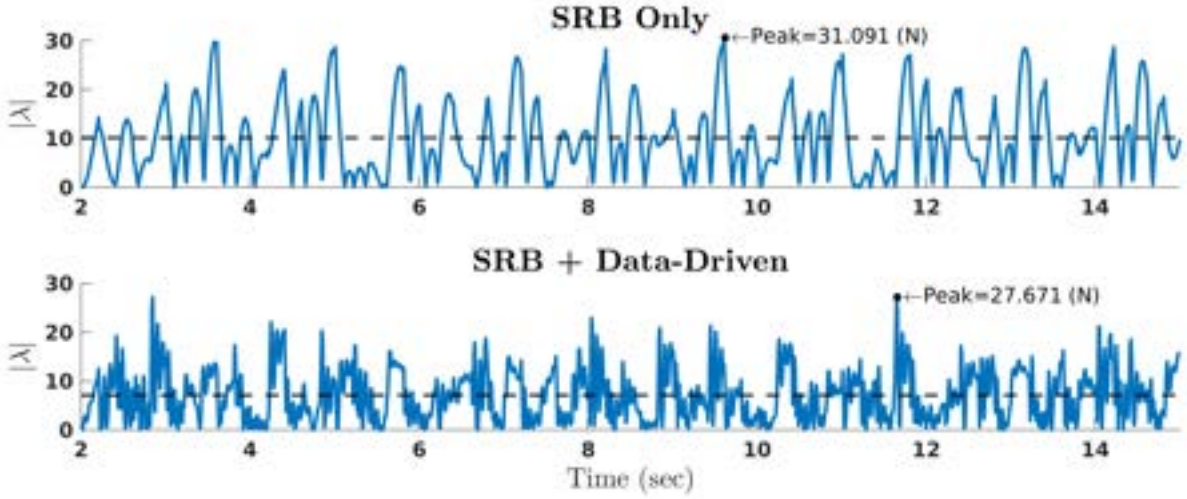

Figure 5.5: This figure shows $|\lambda|$ when using the standard model-based approach (top) and the proposed approach (bottom) for agent 1. The horizontal line indicates the average of $|\lambda|$ for each respective case. It can be observed that both the peak and average are reduced when using the proposed approach.

considerably. Finally, each agent considers the following parameters for the proposed planner: $Q = \text{diag}(1e4, 1e4, 8e5, 1e3, 1e3, 1e3, 1e4, 1e5, 1e3, 1, 1, 1)$ and $R = I \otimes \text{diag}(0.1, 0.1, 0.01)$, where I is an identity matrix of an appropriate size, and $\otimes$ represents the Kronecker product.

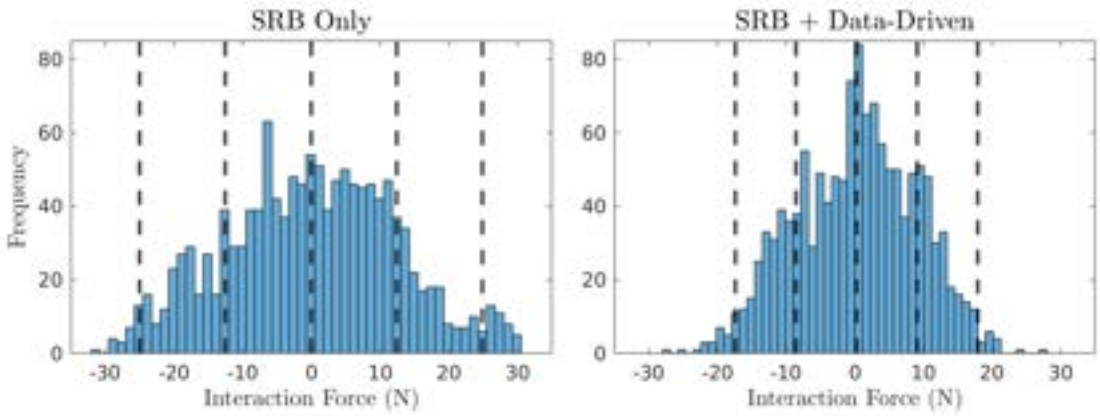

Figure 5.6: This figure shows the distribution of the interaction forces when subject to a periodic disturbance over 15 seconds and corresponds to the simulation in Fig. 5.5. The left plot shows the interaction forces produced when using the purely decentralized MPC, and the plot on the right shows the results when using the proposed distributed MPC that considers the approximate interaction forces. The vertical lines represent the average (center lines) and standard deviations. It is evident that the proposed algorithm produces lower interaction forces more consistently, and has a tighter distribution around zero.

In order to examine the effect of the interaction forces on locomotion and how the proposed approach compares to the strictly model-based approach, the next set of simulations subjects both approaches to an external force of the form $\text{ExtFrc}_1 := \text{col}\,(20\sin(2.6\,t), 30\sin(4.7124\,t), 0)$ in the x, y, and z directions, respectively, and is applied on the COM of one agent. The plots displaying $|\lambda|$ for both the proposed and model-based approach subject to this external force are provided in Fig. 5.5. Although the proposed approach leads to more oscillatory interaction forces, the mean absolute value of the interaction force is reduced by approximately 30%, while the peak force decreases by 11%. Furthermore, the distributions of the interaction forces for the same simulations are provided in Fig. 5.6. This demonstrates that the interaction forces using the proposed approach tend to produce lower interaction forces more consistently, leading to a smaller standard deviation from the mean (which is approximately zero in both cases). The external force applied in this simulation is significant. However, in order to push the algorithms to their limit, an additional simulation was conducted with an even larger external force placed on the second agent as well of the form $\text{ExtFrc}_2 := \text{col}\,(50\sin(12.5512\,t), 25\sin(14.7418\,t), 0)$. In this scenario, the proposed approach still realized about a 19% improvement on the average $|\lambda|$ over the purely model-based approach. In addition, the peak force is reduced by approximately 10%. The improvement is smaller compared to the previous scenario, but this is to be expected as the system is being pushed to the limits.

Figure 5.7: From left to right, this figure shows snapshots of locomotion subject to unstructured terrain, a 9 kg payload (75% of the mass of one agent), and extensive pushes.

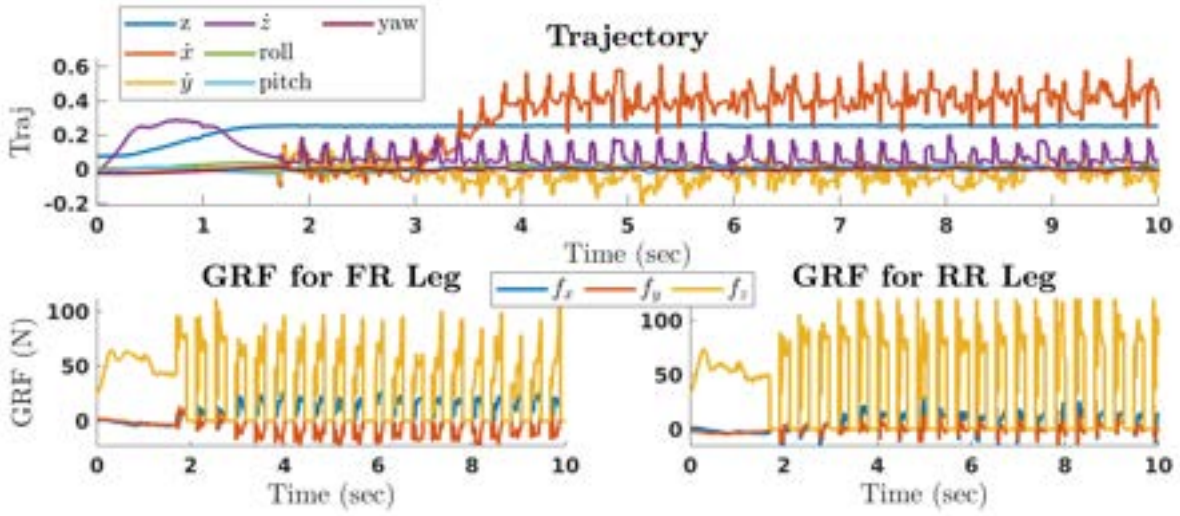

Figure 5.8: This figure corresponds to a nominal hardware experiment (locomotion over flat, unobstructed terrain) and shows the trajectory produced by the proposed MPC (top), the prescribed GRF for the front right leg (bottom left), and the prescribed GRF for the rear right leg (bottom left) for agent 1. Each agent aims to track a forward velocity of 0.5 (m/s) and a standing height of 0.26 (m). The agents also aim to track -0.1 (m/s) in the lateral direction to combat drift.

5.5.3 Hardware Results

Finally, this section provides experimental validation of the proposed approach on hardware. Here, we consider two quadrupeds constrained through a ball joint, similar to the simulations. Analogous to the previous chapters, the optimal trajectory and force profile produced by the planner are passed to the low-level torque controller for tracking. The gains considered for the low-level controller are 1 for torque, $1e4$ for force tracking, and $1e6$ for virtual constraint tracking (trajectory tracking). When using hardware, it is difficult to increase the force gain due to the compliance at the foot. When simulating the full-order system, the force gain can be increased, and the gains on virtual constraints can be lowered, which is often desirable. This is especially the case when considering sub-optimal estimation techniques for position

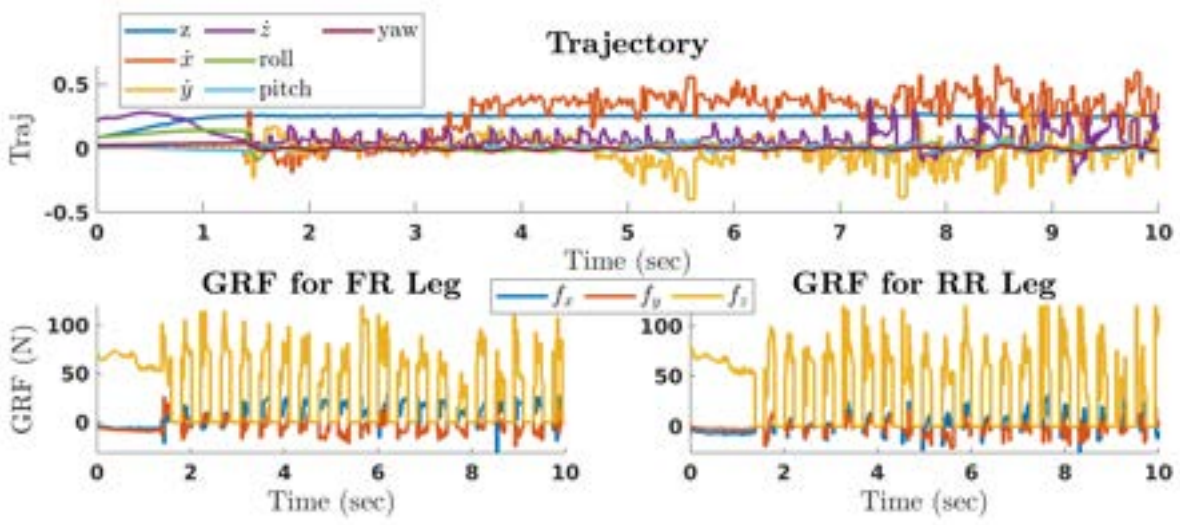

Figure 5.9: This figure shows an experiment of two constrained robots maneuvering over rough terrain composed of randomly placed wooden boards on the ground. The top plot corresponds to the desired trajectory produced by the planner, and the optimal GRFs for the front right and rear right lefts are shown in the bottom left and right, respectively. Each agent aims to track a forward velocity of 0.5 (m/s) and a standing height of 0.26 (m). The data shown here corresponds to agent 1, and the second agent shows similar trends.

and velocity of the COM, as is done in this work. Namely, a simple kinematic estimator is used. The noise from the COM estimate propagates into the control law and leads to noisy torques. However, providing a good force profile to the low-level controller, even when using lower gains, helps with attenuation.

When used in tandem with the low-level controller, we use slightly different gains for the planner than that which was used during reduced-order simulations. In particular, we have $Q = \text{diag}(5e4, 5e4, 8e5, 1e4, 1e4, 1e3, 1e4, 1e4, 1e4, 1, 1, 1)$ and $R = I \otimes \text{diag}(0.1, 0.1, 0.01)$, where I is an identity matrix of an appropriate size, and $\otimes$ represents the Kronecker product. Snapshots of several experiments can be found in Fig. 5.7, showing the team of robots subject to various disturbances, including payloads, unstructured terrain, and external forces. In Fig. 5.8, the nominal trajectory and the optimal GRFs produced by the proposed planner are provided. Although the trajectory is slightly oscillatory, the resulting motion of the system, when used in tandem with the low-level torque controller, is smooth. Furthermore, it is important to note that the GRF profile follows a more ideal trend than that which was presented in Chapter 4. However, this can likely be attributed to the fact that the planner used in this chapter is updated 4 times faster. Furthermore, the force profile of the planner proposed in this chapter follows a more ideal shape, and results in smoother locomotion. In order to examine the system when subject to rough terrain, the team of robots was also subject to unstructured wooden blocks that were randomly scattered on the ground. Similar to the nominal case, the optimal trajectory and GRF produced by the planner can be found

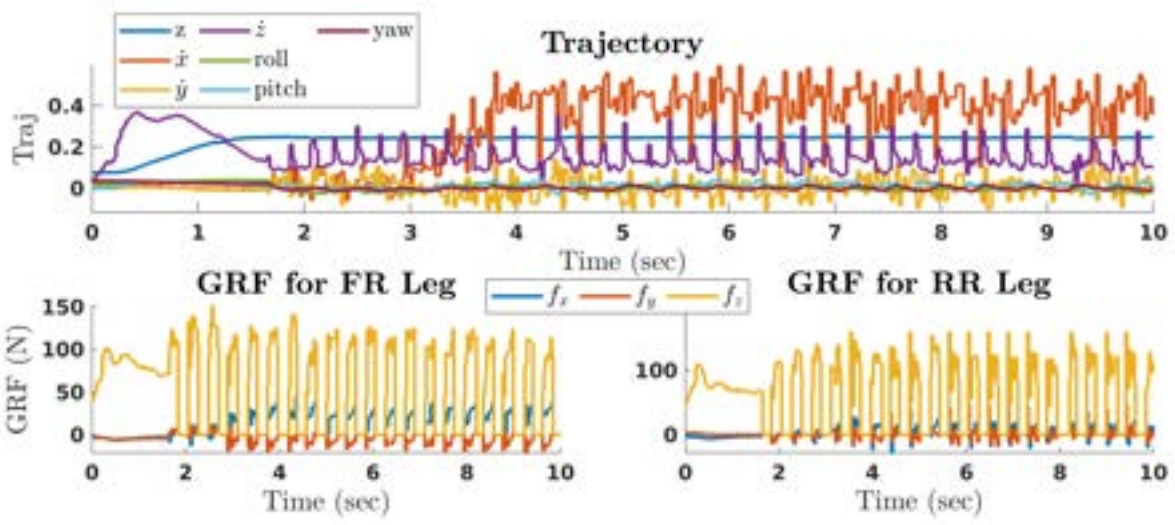

Figure 5.10: This figure shows an experiment of two constrained robots walking subject to a 9 kg payload, which is approximately 72% of the total mass of one agent. The top plot corresponds to the desired trajectory produced by the planner, and the optimal GRFs for the front right and rear right lefts are shown in the bottom left and right, respectively. Each agent aims to track a forward velocity of 0.5 (m/s). The data shown here corresponds to agent 1, and the second agent shows similar trends.

in Fig. 5.9. Even when subject to such terrains, the team of robots is able to maneuver in a robust manner. As a final test for robustness, the team of robots is subject to a 9 (kg) payload, and the results of the planner can be found in Fig. 5.10. This payload corresponds to approximately 72% of the mass of a single agent and is completely unknown to the planner and low-level controller. Although it is difficult to tell from the plot, the resulting standing height only reached approximately 24.5 (cm), while the commanded height was 26 (cm). This offset in the height tracking is also reflected by the positive vertical velocity command given by the planner in an attempt to raise the height. However, the network of robots still maneuvers successfully. In addition to robustness testing, it is important to examine how the system performs when using a time-varying input, as this would be how it is used in practice. For this reason, we finally provide a plot displaying the trajectory from the planner as well as the desired velocity (obtained from a joystick) and actual velocity in Fig. 5.11. It can be observed that the planner is able to track the joystick command with a slight offset.

5.6 Summary

This chapter presented a manner in which physics- and data-based methods can be used in tandem to address holonomically constrained multi-agent systems. In particular, behavioral systems theory is leveraged to parameterize the interaction forces induced by the rigid con-

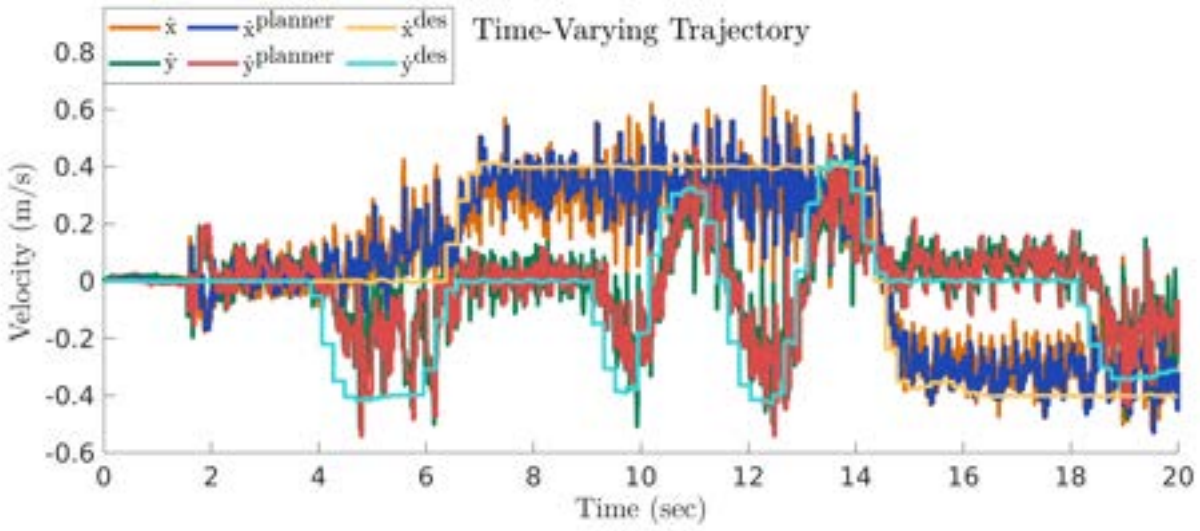

Figure 5.11: This figure shows an experiment of two constrained robots tracking joystick commands, where the data corresponds to agent 1. The commanded joystick velocity ($(\cdot)^{\text{des}}$), planner velocity ($(\cdot)^{\text{planner}}$), and actual velocity are all provided.

straint. At the same time, the local dynamics of each agent are described by the well-known SRB model. When combined with a predictive control framework, the resulting interaction forces are reduced compared to using a physics-based model alone. In particular, the reaction forces are reduced by upwards of 30% on average when subject to external disturbances, which is a considerable improvement. The attenuation of the interaction forces is further complimented by the fact that the proposed method does not change the computation time or sparsity of the predictive controller. The proposed method was also demonstrated to be robust during hardware experiments, where the full hierarchical control scheme is utilized. Namely, an optimal trajectory and force profile is determined in a distributed manner using a high-level controller based on the proposed model, and the result is tracked using a nonlinear whole-body controller at the lower level. The system was subject to various unknown terrains and disturbances and was able to maneuver in a stable manner nonetheless. In addition, the gait produced using the proposed method is much smoother than using just the low-level controller or a purely data-driven method for planning. This is likely due to the improved force profile that is provided to the low-level controller.

Chapter 6

Conclusions and Future Directions

This dissertation details hierarchical control architectures based on nonlinear control techniques and behavioral systems theory in order to obtain stable and robust quadrupedal locomotion. The purpose of this chapter is to provide a summary of this work, followed by future directions of research and concluding remarks.

6.1 Dissertation Contributions

6.1.1 Virtual Constraint Based Torque Controller

We systematically created a low-level controller to perform torque control for quadrupedal locomotion. However, the formulation can be generalized and could then be applied to many other robotic systems with relative ease. This methodology was based on virtual constraints and partial feedback linearization to control the COM of the quadruped. More specifically, we proposed an optimal control law based on a strictly convex QP, subject to feasibility constraints such as the torque limits and friction cone. In addition, we provide a relaxation on the partial feedback linearization subject to a CLF in order to obtain rigorous guarantees in terms of stability, convergence, and robustness. This formulation was examined analytically, resulting in the following two results; 1) owing to the structure of the strictly convex QP and the continuity of the dynamics, we showed that the solution to the QP is continuous with respect to the state in some neighborhood of the periodic orbit, which is essential when leveraging Poincaré techniques, and 2) the orbit is invariant under the flow of the closed-loop dynamics, and the proposed control law is ISS on some neighborhood of the orbit. This control law was further validated in both simulation and on hardware for a trotting gait when subject to various unknown uncertainties, including pushes and rough terrain, and resulted in stable locomotion. Furthermore, it was shown that the CLF remained positive, while its derivative remained negative for the vast majority of the experiments, indicating that rapid exponential stability was achieved, even in the experimental case.

Although excellent results were obtained in both simulation and on hardware, the trajectory provided to the low-level controller was heuristically chosen. In doing so, we obtain sub-optimal motions that can occasionally be jittery or oscillatory. This is also, in part, due to the aggressive or greedy nature of the low-level controller, and the lack of consideration for

future states. This naturally indicates the need for a higher-level trajectory planner such that an optimal trajectory can be provided to the controller proposed here.

6.1.2 Data-Driven Reduced-Order Models

We designed a new trajectory planner that was based on behavioral systems theory; a data-driven method. In particular, this method was used to, in some sense, recreate the SRB model using a direct data-based model. This model was then cast into the form of a convex QP to serve as a trajectory planner for a single quadruped, where the optimal solution was passed to the low-level controller proposed in Chapter 2 for tracking. Instead of using the DeePC approach as has been done in other works, we consider an approximation, which considerably reduced the number of decision variables. In turn, this reduced the computation time, yielding a computationally tractable real-time planner. This planner was further validated both in simulation and on hardware to display the efficacy of the results. In particular, when using this hierarchical control scheme, the robot was able to robustly maneuver over rough and unstructured terrain, including loose wooden blocks, gravel, mulch, grass, and substantial weeds. Furthermore, the robot was able to sustain payloads and could resist external pushes and pulls.

This controller was finally evaluated in three additional ways. Specifically, this work examined 1) how the controller performs when considering different locomotion parameters without collecting new data, 2) how this method compared to the SRB for which it was meant to emulate, and 3) whether or not the trajectory planner could be used with a different low-level controller in a stable manner. In doing so, it was demonstrated that, without collecting additional data for a new model, the system was able to tolerate both a 25% increase and decrease in the stance time for the trot gait (160 (ms) and 250 (ms), respectively). In addition, a walk gait was found to be feasible with the trotting model as well. In terms of the comparison between the traditional SRB model, it was shown that the proposed method was able to produce comparable trajectories. Finally, in order to determine whether or not the planner was agnostic to the choice of low-level controllers, we used the trajectory planner in tandem with the built-in low-level controller provided by the manufacturer, where it was observed that it works but is less robust. It was determined that this was due to the fact that, in the case of using the manufacturers' low-level controller, the planner ran in open-loop since the manufacturer restricts access to some pertinent information.

6.1.3 Data-Driven Multi-Agent Collaborative Locomotion

We proposed an extension to the data-driven method considered in the single agent case to holonomically constrained multi-agent collaborative legged locomotion. To this end, we employed a model based on behavioral systems theory applied to the sophisticated and high-dimensional structure induced by the holonomic constraints for legged robots. The

reduced-order model is formulated in a centralized manner so as to consider the graph induced by the holonomic constraints such that it is encapsulated in the resulting model. The aforementioned model was then used in tandem with distributed predictive control techniques to consider future states while also allowing for the computational demand to be shared across agents. Using a one-step communication delay when distributing allows the algorithm to consider coupling terms between agents with a slight delay, although it was shown that this is a reasonable approach and resulted in robust locomotion. This method was tested on hardware in a variety of scenarios, similar to the single-agent case. Namely, stable locomotion was shown when under the influence of pushes, pulls, and payloads, as well as rough terrain and various outdoor environments. Finally, the scalability of this approach was also examined. In particular, a compound simulation with five agents was provided that shows the agents collectively transporting a large payload while also maneuvering over wavering terrain.

6.1.4 Combining Physics- and Data-Based Methods

The final portion of this dissertation detailed how traditional models can be integrated with data-based models to capitalize on their individual strengths for multi-agent collaborative locomotion. In particular, the SRB model has proven to be a very good reduced-order parameterization for legged robots, while the interaction forces between robots are very complex. For this reason, it was proposed that the two methods be combined such that we have the SRB model subject to interaction forces, where a behavioral model is used to predict the magnitude of the interaction force between agents. Applying this methodology to an interconnected reduced-order system of two agents, it was shown that the interaction forces can be reduced by upwards of 30% in some scenarios, while the peak force is reduced by up to 11%. However, the interaction forces under the proposed approach exhibit more oscillatory behavior, which could be detrimental in some scenarios. Conversely, the reduction in interaction forces is an important step toward being able to robustly carry objects that are fragile since large forces could cause damage. This approach was further evaluated in tandem with the low-level controller in a hierarchical framework such that the proposed algorithm could be tested on hardware. Similar to the other chapters, various robustness tests were conducted, including subjecting the system to external disturbances and maneuvering over rough and unstructured terrain. The system under this control approach was able to move in a stable manner across all of the tested conditions, further emphasizing the efficacy of the trajectory planner and overall control architecture.

6.2 Future Directions

There are several interesting extensions at multiple levels. In particular, when considering the low-level controller, the fully data-based planners, and when combining physics and

data-based methods.

Low-Level Controller: The low-level controller was shown to be robust. However, there are a number of ways in which there is room for improvement. In particular, the CLF considered in the low-level controller operates on the output dynamics and does not consider any of the true nonlinear dynamics of the system. One interesting extension would be to reformulate the problem to consider a passivity-based approach for the output dynamics. In doing so, some of the full-order nonlinear dynamics would appear in the output dynamics. Therefore, the CLF that constrains the partial feedback linearization would encapsulate some of the natural dynamics of the system and may be more representative. In addition, passivity-based I-O linearization has been shown to be more robust to parameter uncertainty, particularly pertaining to the mass, which could lead to improved locomotion when carrying unknown payloads.

Purely Data-Driven Trajectory Planning: In Chapters 3 and 4, we discussed the use of a trajectory planner based solely on a reduced-order data-driven model. These models aimed to create a mapping between GRF and some motion of the center of mass. However, since we are interested in a reduced-order model for planning purposes, it would be worth it to explore simpler input-output pairs. This could be, for example, emulating a double integrator model instead of the SRB model, which would aid in reducing computation time while still potentially describing the dominant dynamics. Conversely, more complex pairs could be examined, too, such as including the relative position of the feet with respect to the center of mass to better parameterize the torques induced about the COM. There is a trade-off between computational demand and accuracy. Generally exploring this relationship, particularly through the chosen input-output pairs, has the potential to provide extremely valuable insight into data-driven reduced-order models.

More specifically related to the multi-agent setting, one could consider creating a model that is concerned with the motion of the centroid of a group of robots as opposed to creating a centralized model that considers the motion of each agent. In particular, one could create a model where the inputs are the GRF for each agent, while the output is the overall motion of the centroid of the group, as opposed to the proposed formulation where the motion of each agent is considered. This could, in turn, allow one to use the agents to collectively manipulate an object together instead of each agent attempting to accomplish its own goals subject to interaction forces. Finally, the multi-agent case can be extended to consider the addition of non-rigid graphs, including both open graphs and serial graphs. This would allow one to produce time-varying configurations, and to dynamically alter a configuration to accomplish a task, such as reorganizing in a manner that would allow the team of robots to navigate a narrow corridor, or spreading out for greater stability when maneuvering over rough terrain.

As a final note on the purely data-driven case, it would be an interesting line of research to look into model reduction techniques for the data-driven models. The models considered in this work require a large set of data due to the nonlinear nature of the system considered.

In turn, the resulting data-driven model is a fat matrix. The reduction could be done in a variety of ways, and some work has already been done in this context, including through the use of the 2-norm approximation used in this work and through the use of singular value thresholding. However, one could similarly consider methods related to generalized eigenvalues to obtain a close square approximation to the original model [194]. Obtaining a square data-driven model is instrumental in being able to use DeePC in a real-time manner, as this would considerably reduce the number of decision variables.

Extending the Combination of Physics- and Data-Based Models: When considering the combination of physics- and data-based models, the most evident direction for future research is extending the method to a greater number of agents. This would be particularly interesting in terms of examining how scalable the approach is, and how the connection graph influences the scalability. Additionally, some of the same extensions apply that were discussed for the purely data-driven multi-agent case as well. In particular, a variety of graphs could be examined. In some sense, this method seems more amenable to various configurations since this method is concerned with approximating the interaction forces directly and is independent of the direction in which the force is applied, so long as the direction can be reconstructed based on the relative position of the agents. Finally, although this method was used for multi-agent collaboration, it is also possible that it could be used to improve a model for a single agent. For example, similar to [61], one may be able to model the discrepancy between the reduced- and full-order models using BST to improve overall performance.

6.3 Concluding Remarks

This work presented a hierarchical control scheme that utilizes data-driven methods to control teams of constrained legged robots. This marks an important step in robot-robot collaboration for legged locomotion, and this work can be extended in a multitude of ways such that future generations of robots can systematically work together to accomplish tasks. We provided several new formulations and applications of behavioral systems theory and followed through with both extensive simulation and experimental results to show the efficacy of the approaches on real quadrupedal platforms.

Bibliography

[1] Boston Dynamics. Spot Robot, https://www.bostondynamics.com/spot, 2021.

[2] MIT news: Mini cheetah is the first four-legged robot to do a backflip, https://news.mit.edu/2019/mit-mini-cheetah-first-four-legged-robot-to-backflip-0304.

[3] Unitree. A1 Robot, https://www.unitree.com/products/a1/, 2021.

[4] 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*, pages 2245–2252, Oct 2018.

[5] Ghost Robotics. Vision 60 Robot, https://www.ghostrobotics.io/partners, 2021.

[6] Thiago Boaventura, Claudio Semini, Jonas Buchli, Marco Frigerio, Michele Focchi, and Darwin G Caldwell. Dynamic torque control of a hydraulic quadruped robot. In *2012 IEEE international conference on robotics and automation*, pages 1889–1894. IEEE, 2012.

[7] C Dario Bellicoso, Koen Krämer, Markus Stäuble, Dhionis Sako, Fabian Jenelten, Marko Bjelonic, and Marco Hutter. Alma-articulated locomotion and manipulation for a torque-controllable robot. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8477–8483. IEEE, 2019.

[8] Boston Dynamics. RHex Robot, https://www.bostondynamics.com/legacy, 2021.

[9] Robust stabilization of periodic gaits for quadrupedal locomotion via qp-based virtual constraint controllers. [Online]. Available: https://youtu.be/0A7mJvVnSn8.

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

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

[12] J.W. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE Transactions on Automatic Control*, 46(1):51–64, Jan 2001.

[13] C. Chevallereau, J. W. Grizzle, and C.-L. Shih. Asymptotically stable walking of a five-link underactuated 3-D bipedal robot. *IEEE Transactions on Robotics*, 25(1):37–50, Feb 2009.

[14] M.W. Spong, J.K. Holm, and D. Lee. Passivity-based control of bipedal locomotion. *IEEE Robotics Automation Magazine*, 14(2):30–40, June 2007.

[15] H. Dai and R. Tedrake. $\mathscr{L}_2$-gain optimization for robust bipedal walking on unknown terrain. In *IEEE International Conference on Robotics and Automation*, pages 3116–3123, May 2013.

[16] G. Song and M. Zefran. Underactuated dynamic three-dimensional bipedal walking. In *Robotics and Automation. Proceedings IEEE International Conference on*, pages 854–859, May 2006.

[17] R.D. Gregg and L. Righetti. Controlled reduction with unactuated cyclic variables: Application to 3D bipedal walking with passive yaw rotation. *IEEE Transactions on Automatic Control*, 58(10):2679–2685, Oct 2013.

[18] K. Byl and R. Tedrake. Approximate optimal control of the compass gait on rough terrain. In *IEEE International Conference on Robotics and Automation*, pages 1258–1263, May 2008.

[19] K. Akbari Hamed and R. D. Gregg. Decentralized event-based controllers for robust stabilization of hybrid periodic orbits: Application to underactuated 3D bipedal walking. *IEEE Transactions on Automatic Control*, 64(6):2266–2281, June 2019.

[20] K. Akbari Hamed and J.W. Grizzle. Event-based stabilization of periodic orbits for underactuated 3-D bipedal robots with left-right symmetry. *IEEE Transactions on Robotics*, 30(2):365–381, April 2014.

[21] B. Morris and J.W. Grizzle. Hybrid invariant manifolds in systems with impulse effects with application to periodic locomotion in bipedal robots. *IEEE Transactions on Automatic Control*, 54(8):1751–1764, Aug 2009.

[22] I. Poulakakis and J.W. Grizzle. The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper. *IEEE Transactions on Automatic Control*, 54(8):1779–1793, Aug 2009.

[23] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle. Compliant hybrid zero dynamics controller for achieving stable, efficient and fast bipedal walking on MABEL. *The International Journal of Robotics Research*, 30(9):1170–1193, August 2011.

[24] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

[25] A. M. Johnson, S. A. Burden, and D. E. Koditschek. A hybrid systems model for simple manipulation and self-manipulation systems. *The International Journal of Robotics Research*, 35(11):1354–1392, 2016.

[26] S. A. Burden, S. S. Sastry, D. E. Koditschek, and S. Revzen. Event–selected vector field discontinuities yield piecewise–differentiable flows. *SIAM Journal on Applied Dynamical Systems*, 15(2):1227–1267, 2016.

[27] R. Vasudevan. *Hybrid System Identification via Switched System Optimal Control for Bipedal Robotic Walking*, pages 635–650. Springer International Publishing, Cham, 2017.

[28] S. Veer, Rakesh, and I. Poulakakis. Input-to-state stability of periodic orbits of systems with impulse effects via poincaré analysis. *arXiv preprint arXiv:1712.03291*, 2017.

[29] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames. Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization. *IEEE Transactions on Robotics*, 34(2):370–387, 2018.

[30] M. Posa, M. Tobenkin, and R. Tedrake. Stability analysis and control of rigid-body systems with impacts and friction. *IEEE Transactions on Automatic Control*, 61(6):1423–1437, June 2016.

[31] W.M. Haddad, V. Chellaboina, and S.G. Nersesov. *Impulsive and Hybrid Dynamical Systems: Stability, Dissipativity, and Control*. Princeton University Press, July 2006.

[32] R. Goebel, R.G. Sanfelice, and A.R. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, March 2012.

[33] Y. Hurmuzlu and D. B. Marghitu. Rigid body collisions of planar kinematic chains with multiple contact points. *The International Journal of Robotics Research*, 13(1):82–92, 1994.

[34] M.W. Spong and F. Bullo. Controlled symmetries and passive walking. *IEEE Transactions on Automatic Control*, 50(7):1025–1031, July 2005.

[35] A. D. Ames, R. D. Gregg, E. D. B. Wendel, and S. Sastry. On the geometric reduction of controlled three-dimensional bipedal robotic walkers. In *Lagrangian and Hamiltonian Methods for Nonlinear Control 2006*, pages 183–196, Berlin, Heidelberg, 2007. Springer.

[36] R. D. Gregg and M. W. Spong. Reduction-based control of three-dimensional bipedal walking robots. *The International Journal of Robotics Research*, 29(6):680–702, May 2010.

[37] I.R. Manchester, U. Mettin, F. Iida, and R. Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, 2011.

[38] A.S. Shiriaev, L.B. Freidovich, and S.V. Gusev. Transverse linearization for controlled mechanical systems with several passive degrees of freedom. *IEEE Transactions on Automatic Control*, 55(4):893–906, April 2010.

[39] E.R. Westervelt, J.W. Grizzle, and D.E. Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(1):42–56, Jan 2003.

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

[41] A.D. Ames, K. Galloway, K. Sreenath, and J.W. Grizzle. Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, April 2014.

[42] K. Akbari Hamed and J. W. Grizzle. Reduced-order framework for exponential stabilization of periodic orbits on parameterized hybrid zero dynamics manifolds: Application to bipedal locomotion. *Nonlinear Analysis: Hybrid Systems*, 25:227–245, August 2017.

[43] Randall T Fawcett, Abhishek Pandala, Aaron D Ames, and Kaveh Akbari Hamed. Robust stabilization of periodic gaits for quadrupedal locomotion via QP-based virtual constraint controllers. *IEEE Control Systems Letters*, 6:1736–1741, 2021.

[44] Jenna Reher, Claudia Kann, and Aaron D Ames. An inverse dynamics approach to control lyapunov functions. In *2020 American Control Conference (ACC)*, pages 2444–2451. IEEE, 2020.

[45] Hamid Sadeghian, Christian Ott, Gianluca Garofalo, and Gordon Cheng. Passivity-based control of underactuated biped robots within hybrid zero dynamics approach. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4096–4101. IEEE, 2017.

[46] C Dario Bellicoso, Christian Gehring, Jemin Hwangbo, Péter Fankhauser, and Marco Hutter. Perception-less terrain adaptation through whole body control and hierarchical optimization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 558–564. IEEE, 2016.

[47] C Dario Bellicoso, Fabian Jenelten, Péter Fankhauser, Christian Gehring, Jemin Hwangbo, and Marco Hutter. Dynamic locomotion and whole-body control for quadrupedal robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3359–3365. IEEE, 2017.

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

[49] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E.R. Westervelt, Carlos Canudas-de Wit, and J.W. Grizzle. RABBIT: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, Oct 2003.

[50] K. Sreenath, H.-W. Park, I. Poulakakis, and J.W. Grizzle. Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL. *The International Journal of Robotics Research*, 32(3):324–345, 2013.

[51] A. E. Martin, D. C. Post, and J. P. Schmiedeler. The effects of foot geometric properties on the gait of planar bipeds walking under HZD-based control. *The International Journal of Robotics Research*, 33(12):1530–1543, 2014.

[52] A. Ramezani, J.W. Hurst, K. Akbai Hamed, and J.W. Grizzle. Performance analysis and feedback control of ATRIAS, a three-dimensional bipedal robot. *ASME Journal of Dynamic Systems, Measurement, and Control December*, 136(2), December 2013.

[53] K. Akbari Hamed, B.G. Buss, and J.W. Grizzle. Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations. *The International Journal of Robotics Research*, 35(8):977–999, 2016.

[54] B. Morris, E.R. Westervelt, C. Chevallereau, G. Buche, and J.W. Grizzle. *Achieving Bipedal Running with RABBIT: Six Steps Toward Infinity*, pages 277–297. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[55] K. Akbari Hamed, W. Ma, and A. D. Ames. Dynamically stable 3D quadrupedal walking with multi-domain hybrid system models and virtual constraint controllers. In *2019 American Control Conference (ACC)*, pages 4588–4595, July 2019.

[56] Q. Cao and I. Poulakakis. Quadrupedal running with a flexible torso: control and speed transitions with sums-of-squares verification. *Artificial Life and Robotics*, 21(4):384–392, Dec 2016.

[57] 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*, 5(3):4463–4470, 2020.

[58] K. Akbari Hamed, V. R. Kamidi, A. Pandala, W. Ma, and A. D. Ames. Distributed feedback controllers for stable cooperative locomotion of quadrupedal robots: A virtual constraint approach. In *American Control Conference (ACC)*, pages 5314–5321, 2020.

[59] Qingyu Liu, Xuedong Chen, Bin Han, Zhiwei Luo, and Xin Luo. Virtual Constraint Based Control of Bounding Gait of Quadruped Robots. *Journal of Bionic Engineering*, 14(2):218–231, 2017.

[60] Vinay R Kamidi, Jeeseop Kim, Randall T Fawcett, Aaron D Ames, and Kaveh Akbari Hamed. Distributed quadratic programming-based nonlinear controllers for periodic gaits on legged robots. *IEEE Control Systems Letters*, 6:2509–2514, 2022.

[61] Abhishek Pandala, Randall T Fawcett, Ugo Rosolia, Aaron D Ames, and Kaveh Akbari Hamed. Robust predictive control for quadrupedal locomotion: Learning to close the gap between reduced-and full-order models. *IEEE Robotics and Automation Letters*, 7(3):6622–6629, 2022.

[62] R.D. Gregg and J.W. Sensinger. Towards biomimetic virtual constraint control of a powered prosthetic leg. *IEEE Transactions on Control Systems Technology*, 22(1):246–254, Jan 2014.

[63] Huihua Zhao, Jonathan Horn, Jacob Reher, Victor Paredes, and Aaron D Ames. First steps toward translating robotic walking to prostheses: a nonlinear optimization based control approach. *Autonomous Robots*, pages 1–18, 2016.

[64] A. E. Martin and R. D. Gregg. Stable, robust hybrid zero dynamics control of powered lower-limb prostheses. *IEEE Transactions on Automatic Control*, 62(8):3930–3942, 2017.

[65] D. Quintero, D. J. Villarreal, and R. D. Gregg. Preliminary experiments with a unified controller for a powered knee-ankle prosthetic leg across walking speeds. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5427–5433, 2016.

[66] A. Agrawal, O. Harib, A. Hereid, S. Finet, M. Masselin, L. Praly, A. Ames, K. Sreenath, and J. Grizzle. First steps towards translating HZD control of bipedal robots to decentralized control of exoskeletons. *IEEE Access*, 5:9919–9934, 2017.

[67] T. Gurriet, S. Finet, G. Boeris, A. Duburcq, A. Hereid, O. Harib, M. Masselin, J. Grizzle, and A. D. Ames. Towards restoring locomotion for paraplegics: Realizing dynamically stable walking on exoskeletons. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2804–2811, 2018.

[68] Vahidreza Molazadeh, Zhiyu Sheng, Xuefeng Bao, and Nitin Sharma. A Robust Iterative Learning Switching Controller for following Virtual Constraints: Application to a Hybrid Neuroprosthesis. *IFAC-PapersOnLine*, 51(34):28–33, 2019.

[69] M. Posa, S. Kuindersma, and R. Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1366–1373, May 2016.

[70] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard. A versatile and efficient pattern generator for generalized legged locomotion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3555–3561, May 2016.

[71] M. Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, 2017.

[72] A. Patel, S. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler. Contact-implicit trajectory optimization using orthogonal collocation. *arXiv preprint arXiv:1809.06436*, 2018.

[73] K. Yunt and C. Glocker. Trajectory optimization of mechanical hybrid systems using sumt. In *9th IEEE International Workshop on Advanced Motion Control, 2006.*, pages 665–671, 2006.

[74] K. Yunt and C. Glocker. A combined continuation and penalty method for the determination of optimal hybrid mechanical trajectories. In H. Y. Hu and Edwin Kreuzer, editors, *Iutam Symposium on Dynamics and Control of Nonlinear Systems with Uncertainty*, pages 187–196, Dordrecht, 2007. Springer Netherlands.

[75] Ayonga Hereid and Aaron D. Ames. Frost: Fast robot optimization and simulation toolkit. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, September 2017. IEEE/RSJ.

[76] Wen-Loong Ma, Noel Csomay-Shanklin, Shishir Kolathaya, Kaveh Akbari Hamed, and Aaron D. Ames. Coupled control Lyapunov functions for interconnected systems, with application to quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 6(2):3761–3768, 2021.

[77] J. B. Martin V, V. R. Kamidi, A. Pandala, R. T. Fawcett, and K. Akbari Hamed. Exponentially stabilizing and time-varying virtual constraint controllers for dynamic quadrupedal bounding. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3914–3921, 2020.

[78] Aaron D Ames and Matthew Powell. Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs. In *Control of Cyber-Physical Systems*, pages 219–240. Springer, 2013.

[79] Shishir Kolathaya, Jacob Reher, Ayonga Hereid, , and Aaron D Ames. Input to state stabilizing control Lyapunov functions for robust bipedal robotic locomotion. In *Proceedings of the American Control Conference*, 2018.

[80] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle. Torque saturation in bipedal robotic walking through control Lyapunov function-based quadratic programs. *IEEE Access*, 3:323–332, 2015.

[81] Quan Nguyen and Koushil Sreenath. Optimal robust control for constrained nonlinear hybrid systems with application to bipedal locomotion. In *2016 American Control Conference (ACC)*, pages 4807–4813. IEEE, 2016.

[82] W-L. Ma, N. Csomay-Shanklin, S. Kolathaya, K. Akbari Hamed, and A. D. Ames. Coupled control Lyapunov functions for interconnected systems, with application to quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 6(2):3761–3768, 2021.

[83] 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*, pages 667–673, Sep. 2017.

[84] 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*, pages 200–207, Dec 2006.

[85] 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*, pages 4420–4427, Sep. 2011.

[86] 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*, volume 2, pages 1620–1626, Sep. 2003.

[87] 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*, pages 1–9, Oct 2018.

[88] Y. Ding, A. Pandala, and H. Park. Real-time model predictive control for versatile dynamic motions in quadrupedal robots. In *International Conference on Robotics and Automation*, pages 8484–8490, May 2019.

[89] M. Chignoli and P. M. Wensing. Variational-based optimal control of underactuated balancing for dynamic quadrupeds. *IEEE Access*, 8:49785–49797, 2020.

[90] 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*, 37(4):1154–1171, 2021.

[91] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robotics and Automation Letters*, 3(3):1458–1465, July 2018.

[92] G. Bledt, P. M. Wensing, and S. Kim. Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4102–4109, Sep. 2017.

[93] Salman Faraji, Soha Pouya, Christopher G Atkeson, and Auke Jan Ijspeert. Versatile and robust 3d walking with a simulated humanoid robot (atlas): A model predictive control approach. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1943–1950. IEEE, 2014.

[94] Yapeng Shi, Pengfei Wang, Mantian Li, Xin Wang, Zhenyu Jiang, and Zhibin Li. Model predictive control for motion planning of quadrupedal locomotion. In *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 87–92. IEEE, 2019.

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

[96] P. Holmes, R. J. Full, D. E. Koditschek, and J. Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Review*, 48(2):207–304, May 2006.

[97] 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*, pages 263–272. ACM, 2014.

[98] H. Dai and R. Tedrake. Optimizing robust limit cycles for legged locomotion on unknown terrain. In *IEEE 51st Annual Conference on Decision and Control*, pages 1207–1213, Dec 2012.

[99] M. A. Hopkins, D. W. Hong, and A. Leonessa. Compliant locomotion using whole-body control and divergent component of motion tracking. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5726–5733, May 2015.

[100] M. A. Hopkins, D. W. Hong, and A. Leonessa. Humanoid locomotion on uneven terrain using the time-varying divergent component of motion. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 266–272, Nov 2014.

[101] Johannes Englsberger, Christian Ott, and Alin Albu-Schäffer. Three-dimensional bipedal walking control using divergent component of motion. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2600–2607. IEEE, 2013.

[102] Johannes Englsberger, Christian Ott, and Alin Albu-Schäffer. Three-dimensional bipedal walking control based on divergent component of motion. *Ieee transactions on robotics*, 31(2):355–368, 2015.

[103] J. Kim and K. Akbari Hamed. Cooperative locomotion via supervisory predictive control and distributed nonlinear controllers. *ASME Journal of Dynamic Systems, Measurement, and Control*, 144(3), Dec 2021.

[104] 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*, 143(7), February 2021.

[105] Ruben Grandia, Andrew J. Taylor, Aaron D. Ames, and Marco Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. In *IEEE International Conference on Robotics and Automation*, pages 8352–8358, 2021.

[106] Junjie Shen and Dennis Hong. Convex model predictive control of single rigid body model on so (3) for versatile dynamic legged motions. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[107] Maria Vittoria Minniti, Ruben Grandia, Farbod Farshidian, and Marco Hutter. Adaptive clf-mpc with application to quadrupedal robots. *IEEE Robotics and Automation Letters*, 7(1):565–572, 2021.

[108] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv: 1909.06586*, 2019.

[109] Maidul Islam, Mohamed Okasha, and Erwin Sulaeman. A model predictive control (mpc) approach on unit quaternion orientation based quadrotor for trajectory tracking. *International Journal of Control, Automation and Systems*, 17(11):2819–2832, 2019.

[110] Xiaodong Zhang, Xiaoli Li, Kang Wang, Yanjun Lu, et al. A survey of modelling and identification of quadrotor robot. In *Abstract and Applied Analysis*, volume 2014. Hindawi, 2014.

[111] Octavio Villarreal, Victor Barasuol, Patrick M Wensing, Darwin G Caldwell, and Claudio Semini. MPC-based controller with terrain insight for dynamic legged locomotion. In *IEEE International Conference on Robotics and Automation*, pages 2436–2442. IEEE, 2020.

[112] 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*, pages 6757–6762, Hong Kong, China, June 2014.

[113] Toni Machado, Tiago Malheiro, Sérgio Monteiro, Wolfram Erlhagen, and Estela 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*, pages 3111–3117, 2016.

[114] Seong Ik Han. Prescribed consensus and formation error constrained finite-time sliding mode control for multi-agent mobile robot systems. *IET Control Theory & Applications*, 12(2):282–290, 2018.

[115] Yuxiao Chen, Andrew Singletary, and Aaron D Ames. Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach. *IEEE Control Systems Letters*, 5(1):127–132, 2020.

[116] Koushil Sreenath and Vijay Kumar. Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. In *Robotics: Science and Systems (RSS)*, 2013.

[117] Aditya Hegde and Debasish Ghose. Multi-uav collaborative transportation of payloads with obstacle avoidance. *IEEE Control Systems Letters*, 6:926–931, 2021.

[118] Jad Wehbeh, Shatil Rahman, and Inna Sharf. Distributed model predictive control for uavs collaborative payload transport. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11666–11672. IEEE, 2020.

[119] Xiaohua Wang, Vivek Yadav, and SN Balakrishnan. Cooperative uav formation flying with obstacle/collision avoidance. *IEEE Transactions on control systems technology*, 15(4):672–679, 2007.

[120] Preston Culbertson, Jean-Jacques Slotine, and Mac Schwager. Decentralized adaptive control for collaborative manipulation of rigid bodies. *IEEE Transactions on Robotics*, 37(6):1906–1920, 2021.

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

[122] Andrea Perizzato, Marcello Farina, and Riccardo Scattolini. Formation control and collision avoidance of unicycle robots with distributed predictive control. *IFAC-PapersOnLine*, 48(23):260–265, 2015.

[123] Daravuth Koung, Isabelle Fantoni, Olivier Kermorgant, and Lamia Belouaer. Consensus-based formation control and obstacle avoidance for nonholonomic multi-robot system. In *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 92–97. IEEE, 2020.

[124] Giancarlo Ferrari-Trecate, Luca Galbusera, Marco Pietro Enrico Marciandi, and Riccardo Scattolini. Model predictive control schemes for consensus in multi-agent systems with single-and double-integrator dynamics. *IEEE Transactions on Automatic Control*, 54(11):2560–2572, 2009.

[125] Tao Yang, Ziyang Meng, Dimos V Dimarogonas, and Karl H Johansson. Global consensus for discrete-time multi-agent systems with input saturation constraints. *Automatica*, 50(2):499–506, 2014.

[126] Tamas G Molnar, Ryan K Cosner, Andrew W Singletary, Wyatt Ubellacker, and Aaron D Ames. Model-free safety-critical control for robotic systems. *IEEE Robotics and Automation Letters*, 7(2):944–951, 2021.

[127] Jeeseop Kim, Randall T Fawcett, Vinay R Kamidi, Aaron D Ames, and Kaveh Akbari Hamed. Layered control for cooperative locomotion of two quadrupedal robots: Centralized and distributed approaches. *arXiv preprint arXiv:2211.06913*, 2022.

[128] Bingyi Xia, Hao Luan, Ziqi Zhao, Xuheng Gao, Peijia Xie, Anxing Xiao, Jiankun Wang, and Max Q-H Meng. Collaborative trolley transportation system with autonomous nonholonomic robots. *arXiv preprint arXiv:2303.06624*, 2023.

[129] Gennaro Notomista, Xiaoyi Cai, Junya Yamauchi, and Magnus Egerstedt. Passivity-based decentralized control of multi-robot systems with delays using control barrier functions. In *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 231–237. IEEE, 2019.

[130] Manao Machida and Masumi Ichien. Consensus-based control barrier function for swarm. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8623–8628. IEEE, 2021.

[131] Dongkun Han and Dimitra Panagou. Robust multitask formation control via parametric lyapunov-like barrier functions. *IEEE Transactions on Automatic Control*, 64(11):4439–4453, 2019.

[132] Beatrice Capelli and Lorenzo Sabattini. Connectivity maintenance: Global and optimized approach through control barrier functions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5590–5596. IEEE, 2020.

[133] Luis Guerrero-Bonilla and Vijay Kumar. Realization of r-robust formations in the plane using control barrier functions. *IEEE Control Systems Letters*, 4(2):343–348, 2019.

[134] Jawhar Ghommam, Luis F Luque-Vega, and Maarouf Saad. Distance-based formation control for quadrotors with collision avoidance via lyapunov barrier functions. *International Journal of Aerospace Engineering*, 2020, 2020.

[135] Eduardo G Hernández-Martínez and Eduardo Aranda-Bricaire. *Convergence and collision avoidance in formation control: A survey of the artificial potential functions approach*. INTECH Open Access Publisher Rijeka, Croatia, 2011.

[136] Ya Liu, Panfeng Huang, Fan Zhang, and Yakun Zhao. Distributed formation control using artificial potentials and neural network for constrained multiagent systems. *IEEE Transactions on Control Systems Technology*, 28(2):697–704, 2018.

[137] Jie Zhang, Gang Wang, Shaohua Yue, Yafei Song, Jiayi Liu, and Xiaoqiang Yao. Multi-agent system application in accordance with game theory in bi-directional coordination network model. *Journal of Systems Engineering and Electronics*, 31(2):279–289, 2020.

[138] Raunak P Bhattacharyya, Derek J Phillips, Blake Wulfe, Jeremy Morton, Alex Kuefler, and Mykel J Kochenderfer. Multi-agent imitation learning for driving simulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1534–1539. IEEE, 2018.

[139] Mohammadhosein Hasanbeig. *Multi-agent learning in coverage control games*. PhD thesis, University of Toronto (Canada), 2016.

[140] Quan Yuan, Shuai Li, Chen Wang, and Guangming Xie. Cooperative-competitive game based approach to the local path planning problem of distributed multi-agent systems. In *2020 European Control Conference (ECC)*, pages 680–685. IEEE, 2020.

[141] Yibei Li and Xiaoming Hu. A differential game approach to intrinsic formation control. *Automatica*, 136:110077, 2022.

[142] Liwei Jiang, Felipe Gonzalez, and Aaron McFadyen. Cooperative game theory based multi-uav consensus-based formation control. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 93–99, 2020.

[143] Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In Gita Sukthankar and Juan A. Rodriguez-Aguilar, editors, *Autonomous Agents and Multiagent Systems*, pages 66–83, Cham, 2017. Springer International Publishing.

[144] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. *Multi-agent Reinforcement Learning: An Overview*, pages 183–221. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[145] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*, pages 321–384. Springer International Publishing, Cham, 2021.

[146] Andrew Singletary, Karl Klingebiel, Joseph Bourne, Andrew Browning, Phil Tokumaru, and Aaron Ames. Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8129–8136. IEEE, 2021.

[147] Thomas Berger, Svenja Otto, Timo Reis, and Robert Seifried. Combined open-loop and funnel control for underactuated multibody systems. In *Nonlinear Dynamics*, volume 95, page 1977–1998, 2019.

[148] Xiao Min, Simone Baldi, and Wenwu Yu. Distributed output feedback funnel control for uncertain nonlinear multiagent systems. *IEEE Transactions on Fuzzy Systems*, 30(9):3708–3721, 2022.

[149] Jin Gyu Lee, Stephan Trenn, and Hyungbo Shim. Synchronization with prescribed transient behavior: Heterogeneous multi-agent systems under funnel coupling. *Automatica*, 141:110276, 2022.

[150] Xiao Min, Simone Baldi, Wenwu Yu, and Jinde Cao. Funnel asymptotic tracking of nonlinear multi-agent systems with unmatched uncertainties. *Systems and Control Letters*, 167:105313, 2022.

[151] William B. Dunbar and Richard M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

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

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

[154] Randall T Fawcett, Leila Amanzadeh, Jeeseop Kim, Aaron D Ames, and Kaveh Akbari Hamed. Distributed data-driven predictive control for multi-agent collaborative legged locomotion. *IEEE International Conference on Robotics and Automation*, In Review, September 2022.

[155] Qasim Ali and Sergio Montenegro. Decentralized control for scalable quadcopter formations. *International Journal of Aerospace Engineering*, 2016, 2016.

[156] Javier Alonso-Mora, Eduardo Montijano, Tobias Nägeli, Otmar Hilliges, Mac Schwager, and Daniela Rus. Distributed multi-robot formation control in dynamic environments. *Autonomous Robots*, 43(5):1079–1100, 2019.

[157] Han-Lim Choi, Luc Brunet, and Jonathan P How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4):912–926, 2009.

[158] Tamás Keviczky, Francesco Borrelli, Kingsley Fregene, Datta Godbole, and Gary J Balas. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on control systems technology*, 16(1):19–33, 2007.

[159] John T Feddema, Chris Lewis, and David A Schoenwald. Decentralized control of cooperative robotic vehicles: theory and application. *IEEE Transactions on robotics and automation*, 18(5):852–864, 2002.

[160] Dong Hun Kim, Hua Wang, and Seiichi Shin. Decentralized control of autonomous swarm systems using artificial potential functions: Analytical design guidelines. *Journal of Intelligent and Robotic Systems*, 45(4):369–394, 2006.

[161] Teddy M Cheng, Andrey V Savkin, and Faizan Javed. Decentralized control of a group of mobile robots for deployment in sweep coverage. *Robotics and Autonomous Systems*, 59(7-8):497–507, 2011.

[162] José M Maestre, D Muñoz De La Peña, and Eduardo Fernández Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32(2):153–176, 2011.

[163] Zhong-Sheng Hou and Zhuo Wang. From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235:3–35, 2013.

[164] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), 2020.

[165] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62), 2022.

[166] Maegan Tucker, Noel Csomay-Shanklin, Wen-Loong Ma, and Aaron D Ames. Preference-based learning for user-guided hzd gait generation on bipedal walking robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2804–2810. IEEE, 2021.

[167] Yandong Ji, Bike Zhang, and Koushil Sreenath. Reinforcement learning for collaborative quadrupedal manipulation of a payload over challenging terrain. In *IEEE International Conference on Automation Science and Engineering*, pages 899–904, 2021.

[168] Jan C Willems, Paolo Rapisarda, Ivan Markovsky, and Bart LM De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.

[169] Jan C Willems. From time series to linear system—Part I. Finite dimensional linear time invariant systems. *Automatica*, 22(5):561–580, 1986.

[170] Ivan Markovsky, Jan C Willems, Sabine Van Huffel, and Bart De Moor. *Exact and approximate modeling of linear systems: A behavioral approach*. Society for Industrial and Applied Mathematics, 2006.

[171] Julian Berberich, Johannes Köhler, Matthias A Müller, and Frank Allgöwer. Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 66(4):1702–1717, 2020.

[172] Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control: In the shallows of the DeePC. In *European Control Conference*, pages 307–312, 2019.

[173] Linbin Huang, Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control for grid-connected power converters. In *IEEE Conference on Decision and Control*, pages 8130–8135, 2019.

[174] Jeremy Coulson, John Lygeros, and Florian Dörfler. Distributionally robust chance constrained data-enabled predictive control. *IEEE Transactions on Automatic Control*, 2021.

[175] Henk J Van Waarde, Jaap Eising, Harry L Trentelman, and M Kanat Camlibel. Data informativity: A new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, 65(11):4753–4768, 2020.

[176] Lai Wei, Yitao Yan, and Jie Bao. A data-driven predictive control structure in the behavioral framework. *IFAC-PapersOnLine*, 53(2):152–157, 2020.

[177] Randall T Fawcett, Kereshmeh Afsari, Aaron D Ames, and Kaveh Akbari Hamed. Toward a data-driven template model for quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 7(3):7636–7643, 2022.

[178] Julian Berberich and Frank Allgöwer. A trajectory-based framework for data-driven system analysis and control. In *European Control Conference*, pages 1365–1370, 2020.

[179] Chris Verhoek, Hossam S Abbas, Roland Tóth, and Sofie Haesaert. Data-driven predictive control for linear parameter-varying systems. *IFAC-PapersOnLine*, 54(8):101–108, 2021.

[180] Kiyotsugu Takaba and Jan C Willems. Concatenability of behaviors in hybrid system interconnection. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, volume 1, pages 370–375. IEEE, 2001.

[181] Thomas Moor, Jennifer M Davoren, and Jörg Raisch. Modular supervisory control of a class of hybrid systems in a behavioural framework. In *2001 European Control Conference (ECC)*, pages 870–875. IEEE, 2001.

[182] B. J. Morris, M. J. Powell, and A. D. Ames. Continuity and smoothness properties of nonlinear optimization-based feedback controllers. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 151–158, 2015.

[183] S. Veer, Rakesh, and I. Poulakakis. Input-to-state stability of periodic orbits of systems with impulse effects via Poincaré analysis. *IEEE Transactions on Automatic Control*, 64(11):4583–4598, Nov 2019.

[184] R. T. Fawcett. *Real-Time Planning and Nonlinear Control for Robust Quadrupedal Locomotion with Tails*. MS thesis, Virginia Tech, Advisor: K. Akbari Hamed, 2021, Available Online: https://www.kavehakbarihamed.com/publications (under publications section for 2021), Direct Link: https://7c91a126-a82e-473d-af78-5fcdc5f4feae.filesusr.com/ugd/d9fe13_fba8bc68a95f451da349e63be2601bf1.pdf.

[185] A. V. Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10(1):287–311, 1976.

[186] Chaohong Cai and Andrew R. Teel. Characterizations of input-to-state stability for hybrid systems. *Systems & Control Letters*, 58(1):47–53, 2009.

[187] A. G. Pandala, Y. Ding, and H. Park. qpSWIFT: A real-time sparse quadratic program solver for robotic applications. *IEEE Robotics and Automation Letters*, 4(4):3355–3362, Oct 2019.

[188] J. Hwangbo, J. Lee, and M. Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, April 2018.

[189] Marc H Raibert. *Legged robots that balance*. MIT press, 1986.

[190] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

[191] Sourav Chatterjee. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.

[192] Kaixiang Zhang, Yang Zheng, and Zhaojian Li. Dimension reduction for efficient data-enabled predictive control. *arXiv preprint arXiv:2211.03697*, 2022.

[193] Michele Focchi, Andrea Del Prete, Ioannis Havoutis, Roy Featherstone, Darwin G Caldwell, and Claudio Semini. High-slope terrain locomotion for torque-controlled quadruped robots. *Autonomous Robots*, 41:259–272, 2017.

[194] Gregory Boutry, Michael Elad, Gene H Golub, and Peyman Milanfar. The generalized eigenvalue problem for nonsquare pencils using a minimal perturbation approach. *SIAM Journal on Matrix Analysis and Applications*, 27(2):582–601, 2005.