Real-Time Planning and Nonlinear Control for Robust Quadrupedal Locomotion with Tails

Randall T. Fawcett

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

Master of Science
in
Mechanical Engineering

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

July 1, 2021
Blacksburg, Virginia

Keywords: Quadrupedal Locomotion, Real-Time Planning, Nonlinear Control

Real-Time Planning and Nonlinear Control for Robust Quadrupedal Locomotion with Tails

Randall T. Fawcett

(ABSTRACT)

This thesis aims to address the real-time planning and nonlinear control of quadrupedal locomotion such that the resulting gaits are robust to various kinds of disturbances. Specifically, this work addresses two scenarios. Namely, a quasi-static formulation in which an inertial appendage (i.e., a tail) is used to assist the quadruped in negating external push disturbances, and an agile formulation which is derived in a manner such that an appendage could easily be added in future work to examine the affect of tails on agile and high-speed motions. Initially, this work presents a unified method in which bio-inspired articulated serpentine robotic tails may be integrated with walking robots, specifically quadrupeds, in order to produce stable and highly robust locomotion. The design and analysis of a holonomically constrained 2 degree of freedom (DOF) tail is shown and its accompanying nonlinear dynamic model is presented. The model created is used to develop a hierarchical control scheme which consists of a high-level path planner and a full-order nonlinear low-level controller. The high-level controller is based on model predictive control (MPC) and acts on a linear inverted pendulum (LIP) model which has been extended to include the forces produced by the tail by augmenting the LIP model with linearized tail dynamics. The MPC is used to generate center of mass (COM) and tail trajectories and is subject to the net ground reaction forces of the system, tail shape, and torque saturation of the tail in order to ensure overall feasibility of locomotion. At the lower level, a full-order nonlinear controller is

implemented to track the generated trajectories using quadratic program (QP) based input-output (I-O) feedback linearization which acts on virtual constraints. The analytical results of the proposed approach are verified numerically through simulations using a full-order nonlinear model for the quadrupedal robot, Vision60, augmented with a tail, totaling at 20 DOF. The simulations include a variety of disturbances to show the robustness of the presented hierarchical control scheme. The aforementioned control scheme is then extended in the latter portion of this thesis to achieve more dynamic, agile, and robust locomotion. In particular, we examine the use of a single rigid body model as the template model for the real-time high-level MPC, which is linearized using variational based linearization (VBL) and is solved at 200 Hz as opposed to an event-based manner. The previously defined virtual constraints controller is also extended so as to include a control Lyapunov function (CLF) which contributes to both numerical stability of the QP and aids in stability of the output dynamics. This new hierarchical scheme is validated on the A1 robot, with a total of 18 DOF, through extensive simulations to display agility and robustness to ground height variations and external disturbances. The low-level controller is then further validated through a series of experiments displaying the ability for this algorithm to be readily transferred to hardware platforms.

Real-Time Planning and Nonlinear Control for Robust Quadrupedal Locomotion with Tails

Randall T. Fawcett

(GENERAL AUDIENCE ABSTRACT)

This thesis aims to address the real-time planning and nonlinear control of four legged walking robots such that the resulting gaits are robust to various kinds of disturbances. Initially, this work presents a method in which a robotic tail can be integrated with legged robots to produce very stable walking patterns. A model is subsequently created to develop a multi-layer control scheme which consists of a high-level path planner, based on a reduced-order model and model predictive control techniques, that determines the trajectory for the quadruped and tail, followed by a low-level controller that considers the full-order dynamics of the robot and tail for robust tracking of the planned trajectory. The reduced-order model considered here enforces quasi-static motions which are slow but generally stable. This formulation is validated numerically through extensive full-order simulations of the Vision60 robot. This work then proceeds to develop an agile formulation using a similar multi-layer structure, but uses a reduced-order model which is more amenable to dynamic walking patterns. The low-level controller is also augmented slightly to provide additional robustness and theoretical guarantees. The latter control algorithm is extensively numerically validated in simulation using the A1 robot to show the large increase in robustness compared to the quasi-static formulation. Finally, this work presents experimental validation of the low-level controller formulated in the latter half of this work.

Acknowledgments

I would like to thank my advisor, Dr. Kaveh Akbari Hamed, for providing a huge amount of support throughout this work. I am extremely grateful for the wealth of knowledge that you have passed down to me, and for the fantastic guidance along the way.

I would also like to extend my gratitude to all of my colleagues in the Hybrid Dynamic Systems and Robot Locomotion Lab for many hours of stimulating conversation and collaborations which made this work possible.

Finally, I would like to thank my friends and family, who always provided ample support for which I am incredibly grateful.

Contents

List of Figures

List of Abbreviations

$(\cdot)^{\text{com}}$, $(\cdot)^{\text{cop}}$ Variables corresponding to the center of mass (COM) and center of pressure (COP)

$(\cdot)_\ell$, $(\cdot)_d$ Variables corresponding to the linearized system, variables corresponding to the discretized system

$(\cdot)_r$, $(\cdot)_t$ Variables corresponding to the robot, variables corresponding to the tail

(f, g, w) Vector fields of the affine continuous-time dynamics

$\mathrm{L}_g\mathrm{L}_f, \mathrm{L}_w\mathrm{L}_f, \mathrm{L}_f^2$ Lie derivatives along the continuous-time dynamics

λ_c, λ_t Ground reaction forces, tail wrench

σ, δf State and input of the single rigid body template model

τ_{roll}, τ_{yaw} Torques applied to the roll and yaw actuators of the tail

τ_r, τ_t, τ Inputs to the robot, the tail, and the augmented system

θ_{roll}, θ_{yaw} Internal roll and yaw angles of the tail

$\zeta[k]$, $\mathcal{U}_{\zeta[k]}$ Current domain function, convex hull formed by stance legs

$B_{(\cdot)}$, $J_{b(\cdot)}$ Input distribution matrix, Jacobian matrix at the base of the tail

$D_{(\cdot)}$, $H_{(\cdot)}$ Mass inertia matrix, nonlinear vector (i.e. Coriolis, centrifugal, and gravitational terms)

F_x, F_y, F_z Forces in the x, y, and z directions

F_t, M_t Forces at the base of the tail, moments at the base of the tail

M, n Total number of continuous-time domains, number of continuous-time domains to plan over

M_r, M_p, M_y Moments in the roll, pitch, and yaw directions

N_d, N_c, T_s Number of grid points to plan over, control horizon, time step between grid points

q_r, q_t, z Configuration and state variables for the robot, the tail, and the augmented system

W, $A_{(\cdot)}$, N, d Extended LIP weighting matrix, state matrix, nonlinear vector function, vector of constants

x, u State variables for the extended LIP model including tail dynamics, inputs to the extended LIP model including tail dynamics

x_k^o, $\dot{x}_k^o$, u_k^o State and input variables for the original and extended reduced-order system at time k

x_0, $\dot{x}_0$, u_0 Operating points around which the augmented LIP model is linearized

y, h_0, h_d Holonomic output function, holonomic control variables, desired evolution of controlled variables

Chapter 1

Introduction

1.1 Overview

This thesis aims to address the planning and nonlinear control of quadrupedal locomotion such that the resulting gaits are robust to various kinds of disturbances. In particular, this work investigates two avenues: 1) a quasi-static control formulation which utilizes an inertial appendage to assist in negating external disturbances, and 2) an agile control formulation which is constructed in a manner such that an inertial appendage may be added in future work.

This work begins by providing a method in which bio-inspired inertial appendages (tails) may be applied to quadrupedal robots in order to increase the robustness of locomotion. Here we present the full-order modeling of an articulated tail and proceed to develop a hierarchical control scheme in which the quadruped and tail may be coupled, allowing for a unified control structure. The control scheme operates in two stages; we first utilize an event-based model predictive control (MPC) based high-level controller which handles real-time path planning, followed by a full-order nonlinear low-level controller to track the trajectories provided by the high-level path planner. The high-level controller operates on a novel reduced-order linear inverted pendulum (LIP) model that has been extended to include the forces produced by the tail. This is done so by augmenting the LIP dynamics to include the tail dynamics as well. The path planner generates center of mass (COM) and tail trajectories subject

to the feasibility of net ground reaction forces (GRF), the shape of the tail, and the tail torques. The generated trajectories are then tracked using virtual constraints via a full-order nonlinear quadratic programming (QP) based input-output (I-O) linearization controller. At the lower level, additional constraints are placed in the GRF at each contacting leg end to further address gait feasibility. Finally, a series of full-order nonlinear simulations using a quadruped augmented with a 2 DOF articulated tail, totaling at 20 DOF, are provided to show the analytical results of the proposed approach. These simulations include multiple ground contact models and different kinds of external disturbances to display the robustness of locomotion when using the presented control scheme. It is shown that the quadruped that has been coupled with a tail produces greater robustness of locomotion as compared to a system without a tail using the same hierarchical controller.

In the latter half of this work, we further extend this hierarchical control scheme (without considering the tail) to a different reduced-order model. In particular, we consider the use of a single rigid body (SRB) model which is linearized along the $SO(3)$ manifold using variational based linearization (VBL). In this case, the high-level MPC is still cast as a convex QP allowing for real-time computation, and the MPC is no longer solved in an event-based manner. The SRB-based MPC provides optimal GRFs for all contacting legs in order to track the desired COM position, velocity, and orientation. This provides a particularly interesting situation in that the feasibility of the GRF at each individual leg is considered inherently in the planner. The SRB model can then be extended to include a tail by augmenting the standard model with additional forces which are to be applied by a tail as opposed to all forces being applied by the leg ends, and this template model allows for much more agile locomotion which lends itself better to real-world implementation. In addition to altering the template model utilized by the high-level planner, the nonlinear low-level controller is augmented with a control Lyapunov function (CLF) to further improve the

physical and numerical stability of the tracking controller. Simulations showing the power of this approach are presented using an 18 DOF quadruped, A1, along with initial experiments displaying the effectiveness of this approach on an actual hardware platform.

1.2 Motivation

Legged robots are intended to be used in many of the same ways as many other unmanned ground vehicles (UGV's). This includes being used for search and rescue, chemical detection, maintenance, inspection, and generally any situations that are too dangerous for humans. However, in many scenarios, the terrain has been designed to accommodate legged organisms (i.e., most man-made structures) or the terrain is rough and unstructured which inhibits the use of traditional wheeled vehicles. In these cases, the use of legged machines are of particular importance in that they are able to adequately move over challenging terrains that may be inaccessible to more traditional robots. Consider the simple example of climbing stairs. While there are wheeled robots which can navigate stairs, it is typically done in a slow manner and requires specialized designs to do so. A quadruped on the other hand could navigate the stairs with similar speed, agility, and dexterity that would be expected of an able bodied dog performing the same task. Not all legged robots or control algorithms are created equal though. Throughout this thesis, quadrupedal robots are considered due to their superior stability properties compared to their bipedal counterparts. By utilizing a system which is more stable while maintaining similar agility, the system is better able to traverse complex terrains and assist in various complex tasks.

Although the construction and control of sophisticated legged machines and biomimetic tails are rapidly advancing (see Fig. 1.1), there is a *fundamental gap in the knowledge* of tractable full nonlinear control algorithms for quadrupedal locomotion, as well as feedback control

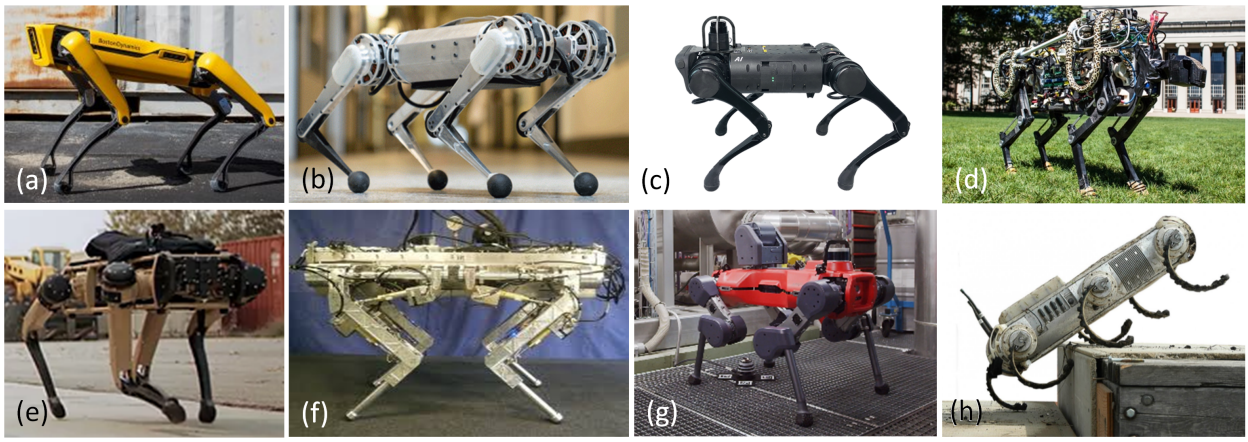

Figure 1.1: (a) Spot (Boston Dynamics) [1] (b) Cheetah Mini (MIT) [2] (c) A1 (Unitree) [3] (d) Cheetah 2 (MIT) [4] (e) Vision 60 (Ghost Robotics) [5] (f) HyQ (Italian Institute of Technology) [6] (g) ANYmal (ANYbotics) [7] (h) RHex (Boston Dynamics) [8].

paradigms that intrinsically couple the movements of tails and legged robots to increase dynamic stability during locomotion. In particular, the following *questions* are still left largely unanswered in the context of quadrupedal locomotion: 1) what are the benefits of coupling *articulated* tails with legged robots in terms of dynamic stability and disturbance rejection? 2) How can path planning be used to effectively coordinate locomotion and tail motion in an online manner? 3) What feedback control paradigms can be used to couple the motion of tails and legged robots for the entirety of the gait cycle? 4) What are the benefits of using a full-order nonlinear low-level controller, and does this lead to highly robust and agile locomotion? and 5) What different high-level path planners can be paired with the nonlinear low-level controller to produce the most robust and agile locomotion while still being amenable to handling additional appendages?

While there are many studies which investigate the use of very simple pendulum tails on legged systems, the addition of articulated appendages—which greatly increases the nonlinearities present—has not been adequately explored and its benefits are not entirely understood, showing a clear disconnect between the design of advanced tails and their accompanying controller. Existing nonlinear legged locomotion controller design approaches are

tailored specifically to systems without the addition of tails, and the integration of tails introduces significant overhead in terms of complexity and dimensionality of these already sophisticated machines. With this increase in dimensionality, the computational burden of both the controller and planning algorithm, which are based on various optimization problems, increases as well. In addition, many investigations of tails consider nominal periodic motions (such as hopping) or static motions, which is not practical for real-world application, further motivating the need to develop advanced planning and control algorithms in order to produce robust gaits for practical scenarios. There have also been many advances in using nonlinear controllers in an online manner for locomotion. However, these methods are often superseded by simplified approaches with many approximations when moving to hardware due to computational demand, which further motivates the development of a low-level controller which is tractable for real world hardware implementations.

1.3 Related Work

1.3.1 Legged Locomotion

Zero Moment Point Locomotion

Initial investigations of legged locomotion, particularly in the case of bipedal locomotion, focused heavily on quasi-static static locomotion due to the inherent increase in stability over its dynamic counterpart. Most early controllers relied on the concept of the Zero-Moment Point (ZMP) criterion in order to produce stable walking [10, 11]. The ZMP is largely a heuristic method which describes the position (projected onto the walking surface) at which all of the individual forces (i.e., the forces provided by each leg individually) may be described as a single equivalent force. The controller then requires that the point at which this force

is acting remains within the convex hull formed by the contacting points of the robot (i.e., the support polygon) with the environment. ZMP based methods have successfully been implemented on various bipedal platforms [12, 13, 14, 15, 16, 17, 18, 19, 20, 21], as well as quadrupedal platforms [22, 23], and still serve as a baseline comparison for many recent works on legged locomotion.

However, this methodology has largely been criticized because of the lack of robustness to unknown environments. When using this control scheme, if the ZMP leaves the support polygon, the system becomes unstable as it loses controllability. The ZMP method also assumes full actuation of the system, and this assumption is violated when the ZMP leaves the support polygon. While some work has been done in attempt to alleviate these issues [24, 25], their use in real world scenarios has been shown to be less effective than those which achieve dynamic stability, as will be discussed further in the forgoing sections. Finally, in all of the cases of bipedal locomotion, the use of ZMP can further complicate the design and control due to the fact that the robot must be actuated at the ankle, and cannot have point feet. In the case of point feet of a biped, the support polygon reduces to a single point during single contact domains, which is highly impractical for real world implementation.

Possibly most importantly, the nature of legged locomotion is unstable and dynamic. While creating a quasi-static controller can produce reasonable results for a variety of systems, the goal of producing natural, robust, adaptable, underactuated, and highly agile legged locomotion requires the use of dynamic path planning and more dynamic control methodologies. However, this heuristic method provides good insight into the stability of a legged robot and can be used in many areas, such as in the development of a quasi-static path planner, and for heuristic foot placement techniques.

General Legged Locomotion Control

More recently, the advent of hybrid systems methods [26, 27] have enabled more agile gaits. Hybrid systems theory aims to break the gait cycle into various continuous-time and discrete-time domains, wherein the continuous-time domains describe the motion of the system using standard Lagrangian dynamics, and the discrete-time domains describe the instantaneous changes in velocity of the system during rigid impacts of any specified number of swing legs with the walking surface [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51]. A variety of state-of-the-art nonlinear controllers exist which utilize hybrid systems to create stable locomotion, including controlled symmetries [31], hybrid reduction [52, 53], transverse linearization [33, 54], and hybrid zero dynamics (HZD) [30, 41, 55, 56, 57]. However, HZD and transverse linearization are the only methods which are capable of handling the general case of underactuation inherent to dynamic locomotion. In HZD methods, the motion of the limbs are coordinated by enforcing a set of kinematic constraints, referred to as virtual constraints [28, 56], which are tracked via I-O feedback linearization [58]. Virtual constraint controllers have been validated both numerically and experimentally for the stable locomotion of bipedal robots [30, 40, 43, 49, 56, 59, 60, 61, 62, 63], quadrupedal robots [64, 65, 66, 67, 68], lower extremity powered prostheses [69, 70, 71, 72], and exoskeletons [73, 74, 75].

One of the barriers to using a virtual constraints controller in the past has been the need for offline trajectory planning in order to create agile gait patterns. The path planning of legged robots is typically generated by solving a nonlinear programming (NLP) problem (see, e.g., [49, 76, 77, 78, 79, 80, 81]) through the use of existing solvers, such as in [49, 82]. While these NLP problems can be solved relatively efficiently in an offline manner, they *cannot* be used in a real-time manner which prevents the use of HZD methods alone when navigating complex environments. Furthermore, HZD methods generally require periodic gait cycles,

often contain restrictions on starting and stopping criteria, and are not often highly robust to unknown environmental changes which restricts its use to known controlled environments. For these reasons, the typical offline planning protocol must be replaced with some form of real-time planning, wherein the trajectory from the planner can then be tracked efficiently with an HZD based virtual constraints controller.

In addition to path planning concerns, there are also some numerical concerns when using I-O linearization. Specifically regarding legged locomotion, there is a possibility that the decoupling matrix becomes ill-conditioned or singular, which severely impacts the controller and can cause the system to produce undefined behavior. In order to alleviate this issue, many researchers have cast the I-O linearization into the form of a convex quadratic program (QP) [30, 66, 67, 83, 84, 85, 86] where a defect variable can be added to the formulation in an attempt to provide the torque that follows the desired outputs as well as possible while enforcing other constraints such as feasible torques and ground reaction forces (GRF). Unfortunately, even though one of the goals of this QP is to reduce the magnitude of the slack variable placed on the I-O linearization, the slack variable changes in an unstructured manner with no regard for the stability of the outputs. Recent advents in legged locomotion have included a control Lyapunov function (CLF) applied to the output dynamics as one of the constrains in the QP to help ensure that manner in which the slack variable changes over time is consistent with rapidly exponentially stabilizing output dynamics [30, 83, 84, 85, 86]. The CLF also allows for better dynamic balance between tasks which may have conflicting goals in the case of adding additional appendages, such as arms or tails [83].

Model Predictive Control

Another class of controllers include a variety of QP- and MPC-based techniques to produce stable and agile locomotion. This includes LIP-based MPC path planning [66, 87, 88, 89, 90],

single rigid body based MPC [91, 92, 93, 94], nonlinear MPC [95], policy-regularized MPC [96], and whole-body QP-based control [97, 98]. Of the methods that are used for control of quadrupeds, LIP and single rigid body techniques have been studied most extensively. During LIP-based MPC, COM and COP trajectories are generated subject to the ZMP criteria and feasible net GRF. This methods does not require the use of an offline path planner, though there is generally a need to couple LIP-based MPC with a secondary nonlinear controller to bridge the gap between the reduce- and full-order model. This technique has generally been explored on bipedal hardware platforms [87, 88, 89], but has recently been extended to quadrupedal systems as well [66]. Being based on the ZMP, the LIP model produces quasi-static locomotion. Due to the fact that quadrupeds have more legs than a biped, the use of a LIP model for locomotion has shown to produce very robust locomotion when moving over flat ground. Unfortunately, the implementation of such LIP model often requires knowledge of footstep locations for at least a couple steps out which can make robust implementation difficult unless a sophisticated footstep planner is in place or footsteps are known in advance.

Alternatively, single rigid body models have successfully been implemented on quadrupeds [91, 92, 93, 94]. There are several advantages to using the single rigid body formulation, including the fact that the GRFs are automatically considering the the planning stage, and that the single rigid body dynamics place no explicit constraints on the COP which allows for more versatile and dynamic motions. However, the single rigid body dynamics are nonlinear due to the rotational dynamics. In order to make the MPC convex, which leads to significant improvements in computational speed, the dynamics are successively linearized. This linearization has been performed in various manners. In initial investigations, the yaw dynamics were linearized successively and the roll and pitch were assumed to be zero [91], but more recently there has been a trend toward the use of variational based linearization

(VBL) approaches such that the dynamics are linearized along the $SO(3)$ manifold resulting in a singularity free representation of the rotational dynamics [92, 93]. The use of VBL techniques allows for planning when the robot is in any configuration, including those which would be considered singular under the standard Euler-angle representation of rotation. These MPC formulations are generally run in at a slower rate ($50 - 500$ Hz) and are used as a path planner. The MPC determines the optimal forces at the contacting leg ends, and the use of a simple force-torque mapping determines the torques corresponding to those forces ($500 - 1000$ Hz), but no information is ascertained regarding the swing leg, therefore necessitating the use of an additional low-level controller to track the motion of the swing leg. While this has worked well in practice for legged robots which have negligible leg and foot mass/inertia relative to the body, alternative techniques are required for robots with greater leg mass/inertia.

1.3.2 Tails

Tail Designs

The use of tail-like appendages has been explored for a variety of purposes, though they have been studied most extensively for use in active reorientation during falls. The design of said tails extends from the most simple pendulum-like structure [99, 100, 101, 102, 103, 104, 105, 106], to multi-DOF serpentine structures [107, 107, 108, 109, 110, 111]. The simple tails listed are directly driven or include a small gear ratio, though [112] provides a particularly interesting manner in which a 2 DOF pendulum tail may be implemented compactly. The serpentine structures listed are cable and link driven which allows for the actuation to be located at the base, keeping the body of the tail relatively light. Having a relatively heavy base while the rest of the tail remains light in comparison also aligns with

many biological organisms with tails [113]. One of the major drawbacks to articulated tails in general is weight which is particularly concerning when mounted on a quadruped due to the fact that the tail weight will reduce the payload capability of the robot when performing tasks. Another distinct difficulty of creating an articulated tail, particularly for cable driven systems, is ensuring uniform deformation along the length of the tail. In some cases, creating uniform deformation may include using several different cables (see e.g., [114]), while in other cases gears can be used to holonomically constrain the system (see e.g., [107]).

Though not inertial appendages, there have been several other unique developments which are applicable here. Recent work has suggested that the best tail designs may not be inertial appendages at all, and suggests that the aerodynamic drag created by specially designed tails could be more advantageous [115]. In particular, there have been studies that show the effectiveness of a cheetah tail based on its aerodynamic properties [116], and these concepts have recently been extended for the use in robotic tails. Although this avenue of research is in its infancy for use on quadrupedal systems, the concept has proven to be effective and warrant's further exploration. Finally, the design of several robotic fingers provide an interesting mechanical design for a driving gear train [117, 118]. These robotic finger designs use gears to create uniform deformation along the length of the finger, but also use them to transmit torque to each joint which removes the need for cable driven solutions when using an inertial based tail. While the application is to grasping mechanisms, the concept easily extends to tail mechanisms as well.

Tail Controllers

The use of inertial limbs for stability has been studied in nature primarily in terms of dynamic reorientation during jumps and falls [113, 115, 119, 120], regaining static stability [121], and during dynamic and agile locomotion [122]. Similar to their biological counterpart,

the use of robotic tails on legged machines has been explored on bipedal [103, 104, 106, 112], quadrupedal [101, 105, 115, 123, 124, 125], hexapedal [102, 126, 127], octapedal [128], and wheeled [129, 130] systems. These studies mainly investigate the use of tails in performing aerial reorientation [102, 103, 104, 105, 106, 112, 115, 124, 125]. This means that the controllers created are able to use models based on the conservation of angular momentum which *cannot* easily be extended to stance phases of locomotion. In turn, many of the listed studies use simple proportional-integral-derivative (PID) controllers which operate on the error between the current and desired orientation of the system. Tails have also been explored to negate external disturbances during quadrupedal contact [101], to enable rapid accelerations during gait initiation [115, 123], for heading control of slip-based legged robots [126, 127, 128], and for stability of small terrestrial vehicles [129, 130]. In these cases the controllers are also based on simple PID control approaches. Finally, the dynamic stability of legged machines augmented with biomimetic tails has been addressed in very few studies (see e.g., [100, 131]). In both of these cases the system is following a trajectory in an open-loop manner, where the trajectory is calculated in an offline manner using a central pattern generator. While this does produce stable locomotion, there is a large shortcoming in the systems ability to negate external disturbances during locomotion due to the nature of both offline path planning and open-loop control.

1.4 Goals, Objectives, and Contributions

*The **overarching goal** of this thesis is to present a formal foundation to design real-time motion planning and nonlinear feedback control algorithms for robust and agile locomotion of quadrupedal robots augmented with serpentine tails.* The **specific objectives** and **key contributions** of the work are as follows (see Fig. 1.2):

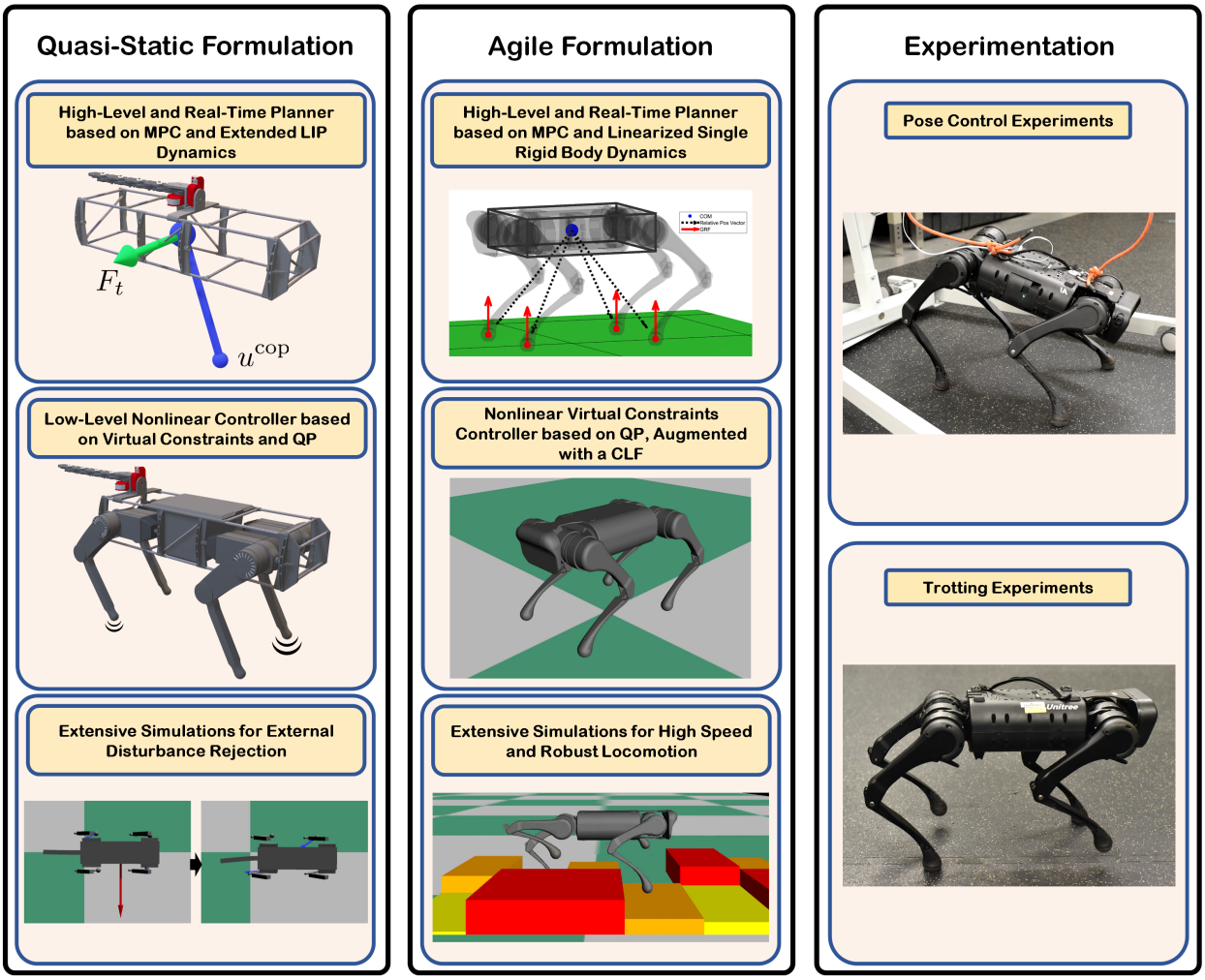

Figure 1.2: Overview of the goals and contributions of this work.

1. A novel reduced-order model, termed the extended LIP model, is presented for the real-time planning of the trajectory of the COM motion and tail motions. Specifically, the presented reduced-order model augments the standard LIP model with the nonlinear tail dynamics to enable us to generate trajectories through an online optimal control problem.

2. A hierarchical nonlinear control scheme for the motion control of quadrupedal robots with robotic tails is presented. At the higher-level of the control scheme, an MPC based formulation is proposed for the real-time planning of the reduced-order model to

optimally generate trajectories for the quadrupeds COM and each actuated tail joints. The feasibility of the motion is directly considered in the MPC to ensure that the tail range of motion, the constraint on the location of the COP, and feasibility of the net GRF are all respected. Due to the nonlinearities of the reduced-order model, the extended LIP model is successively linearized at the beginning of each new continuous-time domain, which results in a convex QP that is solved in an event-based manner (see Fig. 1.3).

3. At the lower-level of the control scheme, a nonlinear controller, based on QP and virtual constraints, is presented for the motion control of the full-order dynamical system to track the prescribed motions of the COM and tail while imposing feasibility of the GRFs at all contact points. In the latter portion of this work, this nonlinear controller is extended slightly to incorporate a control Lyapunov function as an additional constraint on the convex QP. In doing so, the system is guaranteed to converge to the output dynamics in a rapidly exponentially stable manner whenever feasible.

4. We investigate the effect of robotic tails on quadrupedal locomotion intrinsically coupled with the proposed control algorithms. For this purpose, a series of extensive and full-order numerical simulations is presented to demonstrate the effectiveness and robustness of the proposed control approach for locomotion of a 20-DOF advanced quadrupedal robot, Vision 60, augmented with a serpentine tail in the presence of external disturbances (see Fig. 1.4). We numerically show that the integration of the tail dynamics with the developed nonlinear feedback control algorithms can significantly reduce the effect of external disturbances on quadrupedal locomotion.

5. In a similar manner to the aforementioned hierarchical nonlinear control scheme proposed with the extended LIP model, a more dynamic reduced-order model, specifically a single rigid body (SRB) model, is considered as an alternative path planner in order

to generate highly robust and agile locomotion. This model is implemented in a manner which may easily be extended to locomotion with the addition of a tail to further add robustness.

6. A series of simulations of the full order, 18 DOF quadruped, A1 (see Fig. 1.5), are presented in order to demonstrate the robustness and resilience of the developed hierarchical control scheme when using the MPC with the SRB model and the low-level QP augmented with a CLF. In particular, simulations are provided which show high-speed locomotion, locomotion over significant unknown ground height variations, and locomotion subject to various external disturbances.

7. Preliminary experiments on the A1 robot are provided for the latter low-level control formulation wherein the CLF is utilized and heuristic trajectories are provided (i.e., the MPC is not used at this stage of preliminary experimentation). This is the first time in which a full-order and nonlinear HZD controller with exact feedback linearization has been implemented on hardware. We note that the use of a model-based CLF on hardware has only been validated in a select few cases on bipedal platforms, making this the first model-based CLF implementation for quadrupedal locomotion. Experiments include pose control, trotting in place, and forward/backward/lateral trotting.

References [100] and [131] have provided an interesting approach to address the dynamic stability of legged robots augmented with a single-DOF pendulum-like tail, but in both cases the controller used is a central pattern generator (CPG), where the trajectory is calculated offline and the robot is controlled in an open-loop manner. Reference [100] provided an analysis regarding the general design of the tail using a spring loaded inverted pendulum (SLIP) model, though it is not used as part of any control algorithm and only encompasses movement in the sagittal plane. This work differs completely in that, 1) we consider the use of a serpentine tail, 2) we propose real-time planning algorithms for both locomotion

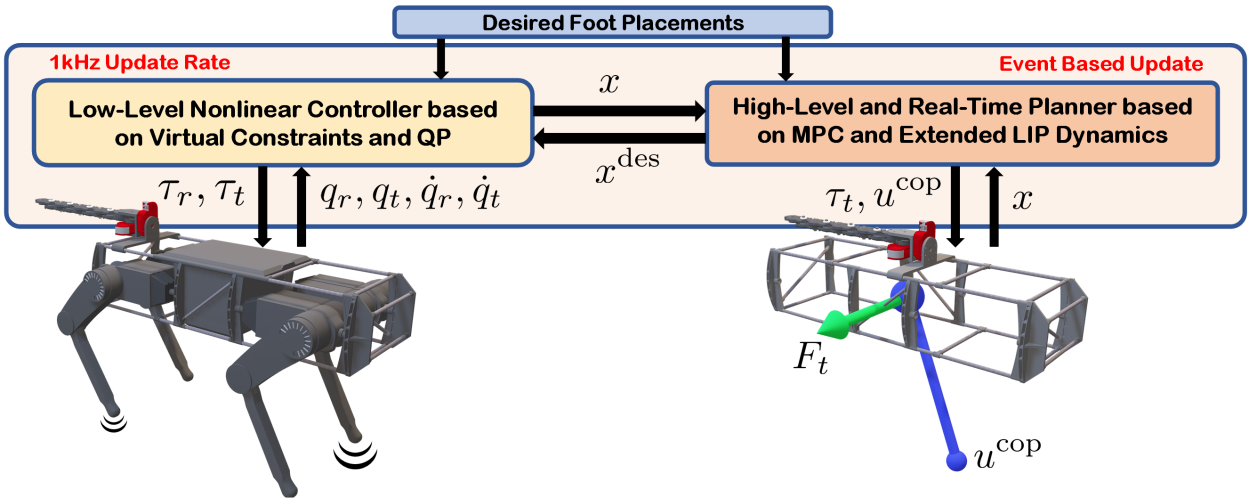

Figure 1.3: Block diagram of the proposed hierarchical control scheme. A high-level and real-time planner, based on event-based MPC, computes the optimal COM and tail trajectories for a reduced-order model, referred to as the extended LIP dynamics, subject to the feasibility of the net GRF and tail motion. A low-level nonlinear controller, based on virtual constraints and QP, imposes the full-order dynamical model of locomotion to track the prescribed and reduced-order optimal trajectories subject to the feasibility of the GRFs at the contacting leg ends.

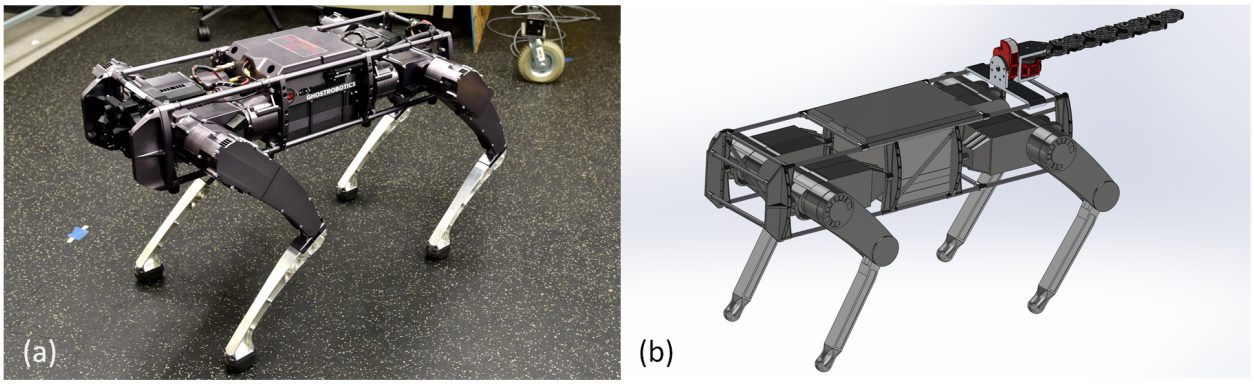

Figure 1.4: (a) Vision 60 robot with 18 DOFs whose full-order dynamical model will be used for the numerical simulations. (b) CAD representation of Vision 60 augmented with a 2-DOF tail mechanism.

and the tail by using an extended LIP model, and 3) we track the generated reduced-order trajectories in a closed-loop manner through a QP-based virtual constraint controller. The work presented in the current paper also differs from the previous work [66] in that [66] did

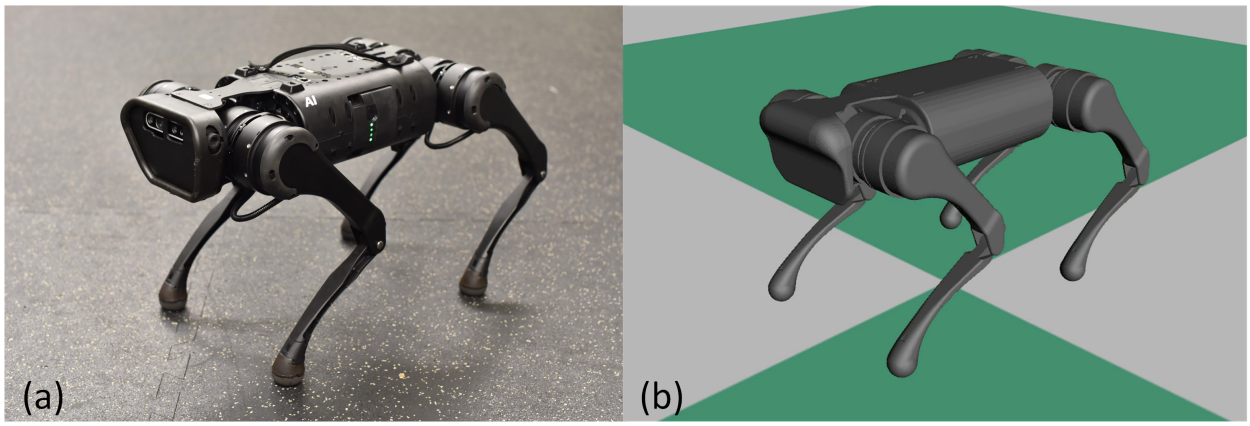

Figure 1.5: (a) A1 robot with 18 DOFs which is used for both the numerical simulations and experiments provided in Chapter 4. (b) Model representation of the A1 robot in RaiSim [9].

not address locomotion with tail dynamics. Here, we derive and use a novel extended LIP model in which additional external forces may be addressed. Furthermore, the current work includes the dynamics of the appendage in the path planner in order to increase gait stability and robustness as opposed to treating the appendage as a disturbance. We also extend the event-based MPC formulation of [66] to the *nonlinear* reduced-order model, arising from the extended LIP dynamics and plan for the optimal COM motion of the robot as well as the tail trajectory.

In terms of the more dynamic single rigid body model discussed later in this work, this model has been used by several groups and has seen successful implementations in both simulation and experimentation [91, 92, 93]. However, the current work stands apart primarily due to the nonlinear low-level controller used to track the forces from the high-level MPC. Furthermore, previous works use *only* the forces produced by the MPC, whereas this formulation uses the forces to condition the QP while also considering the need to track the position and velocity of the COM as prescribed by the MPC. This allows the low-level and full-order nonlinear controller to dynamically balance the reduced-order forces, positions, and velocities, and hence, bridges the gap between reduced- and full-order dynamical models of locomotion

for robots with significant leg mass/inertia.

1.5 Relevant Publication and Contributed Work

As part of this research, a journal article has been written and published as follows:

- **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*, 143(7), 071004, Feb, 2021, https://doi.org/10.1115/1.4049555

Material from Chapters 2 and 3 comprise and extend the material published in this journal article.

1.6 Thesis Outline

This work is organized as follows. Chapter 2 discusses the tail design, tail dynamics, the overall nonlinear system dynamics of quadrupeds, and the coupled dynamics of quadrupeds with tails. Chapter 3 details the proposed extended LIP model, provides the formulation for both the reduced-order path planner and the full-order nonlinear low-level tracking controller, and finally a series of simulations are presented to show the efficacy of the proposed approach. Chapter 4 develops a new framework which extends the previous work to more agile locomotion in a manner which is conducive to the addition of an external appendage, including tails. This chapter contains a series of full-order simulations as well as preliminary experiments to demonstrate the feasibility of this proposed approach. Finally, in Chapter

5, a discussion of the results, a summary of the contributions of this work, and implications and directions for future work are discussed.

Chapter 2

Modeling and Nonlinear Dynamics

This chapter provides an overview of the design of the serpentine tail considered in this work, followed by a detailed derivation of the full-order dynamical models of both the tail, the quadruped, and the coupling between them.

2.1 Tail Design

The tail design is a 2 DOF system consisting of 1 DOF for roll motion and 1 DOF for yaw motion (see Fig. 2.1). The yaw portion of the tail is similar to [107] in that the the yaw gears are holonomically constrained to create uniform deformation along the length of the tail. It is also similar to [118] in that the driving mechanism is the gear train itself. Using the gear train as the driving mechanism greatly simplifies the hardware needed at the base of the tail while also decreasing the design complexity of each link do to the lack of cable routing.

Here we consider links which have two separate herringbone gears, one on each end in opposing directions. Herringbone gears were chosen to assist in gear centering during operation of the tail and to aid in the prevention of drooping along the length of the tail. In order to define how each link is constrained, we define ℓ_n, where n indicates the link number. Let ℓ_0 be the driving gear on the actuator and ℓ_1 be a fixed link rigidly attached to the roll cage. Every subsequent link has the following relationship: the gears of ℓ_n mesh with ℓ_{n-2} and

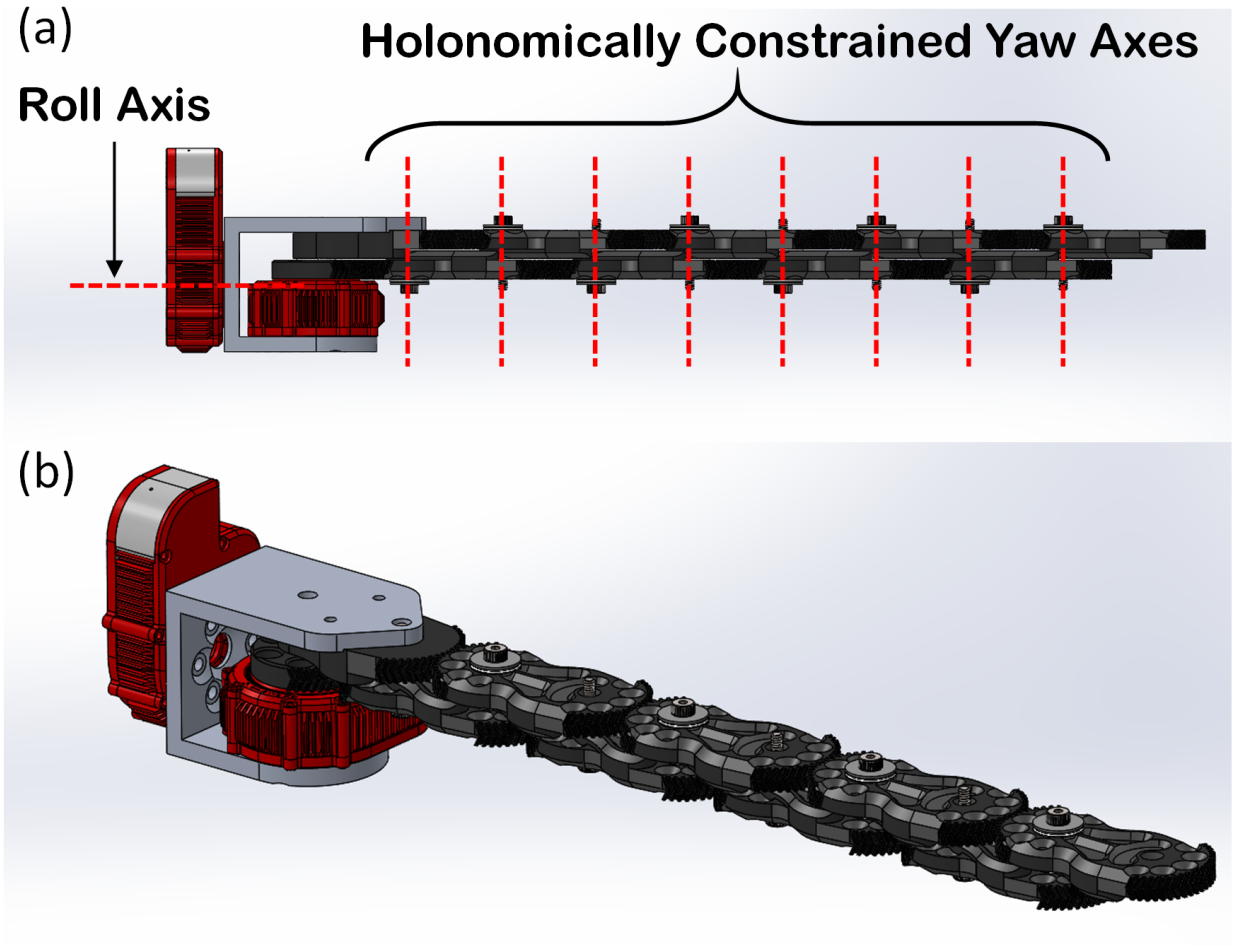

Figure 2.1: (a) Side view of the CAD model of the tail with labeled axes of rotation (b) Isometric view of the tail CAD design

ℓ_{n+2}, and are pinned to ℓ_{n-1} and ℓ_{n+1} (see Fig. 2.2). Since each gear interface has the same gear ratio, ℓ_n will rotate by the same amount as ℓ_{n-1}, aside from ℓ_1 which remains fixed. The set of yaw links connect directly to a roll cage, which is attached directly to the roll actuator. In this case, we consider the use of a Hebi-X5 actuator[1] which includes a hefty moment bearing at its output, allowing for direct connection without additional bearing hardware. In addition to reducing the need for additional bearings, the Hebi-X5 actuator comes with built in electronics that allow for torque control commands to be issued and

[1]https://www.hebirobotics.com/

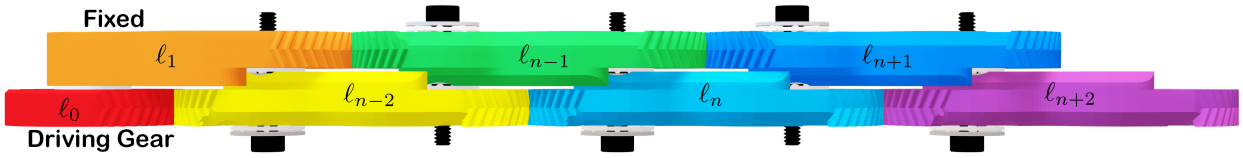

Figure 2.2: Illustration of the construction and meshing of the yaw mechanism of the tail. Here, link ℓ_n is pinned to links ℓ_{n-1} and ℓ_{n+1}, and the gears of link ℓ_n meshes with links ℓ_{n-2} and ℓ_{n+2}.

position/velocity measurements to be read over Ethernet at 1kHz, so no further electronics are necessary to drive the system.

The weight of the tail comes in at approximately 1.51 (kg), in which the base, roll link, and individual yaw links are approximately 0.4 (kg), 0.68 (kg), and 0.053 (kg), respectively. It is worth noting that many of the other articulated tails listed previously weight significantly more than the proposed tail design. For example, the entire tail mechanism considered in [107] weighs 4.02 (kg).

2.2 Full-Order Nonlinear Dynamics

The generalized coordinates for Vision 60[2] (see Fig. 2.3) are expressed as $q_r := \text{col}(p_r, \alpha_r, q_b) \in \mathcal{Q}_r \subset \mathbb{R}^{18}$, in which the subscript "r" stands for the robot, $p_r \in \mathbb{R}^3$ and $\alpha_r \in \mathbb{R}^3$ represent the absolute position and orientation of the body of the robot in the inertial world frame, respectively, and $q_b \in \mathbb{R}^{12}$ denotes the body angles (i.e., shape of the robot). The torque and force inputs applied to the robot are denoted by $u_r \in \mathbb{R}^{12}$, $\lambda_c \in \mathbb{R}^{3n_c}$, and $\lambda_t \in \mathbb{R}^6$, which are the actuator torque inputs, the GRFs at the contacting leg ends, and the tail reaction wrench (forces and moments), respectively. In our notation, "col" represents the column vector and n_c denotes the number of contacting legs with the ground. The evolution of the

[2]https://www.ghostrobotics.io/

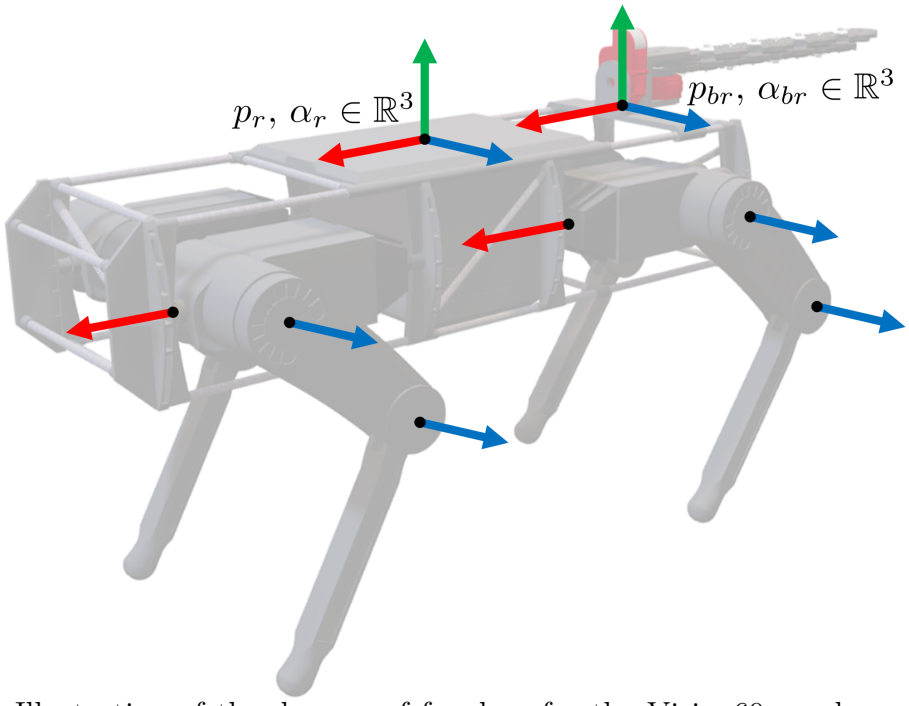

Figure 2.3: Illustration of the degrees of freedom for the Vision60 quadruped, where each arrow indicates a direction of motion and/or an axis of rotation. The body consists of 6 DOF at the center of mass (position and orientation), and each leg contains 3 rotational DOF, including the hip roll, hip pitch, and knee pitch. Finally, the coordinate frame at the coupling of the tail and the body of the robot is represented. This quadruped weighs approximately 32 (kg) and stands around 40 (cm) off the ground.

quadruped is then described by the Euler-Lagrange equations and principle of virtual work as follows:

$$D_r(q_r)\,\ddot{q}_r + H_r\,(q_r, \dot{q}_r) = B_r\,\tau_r + J_c^\top(q_r)\,\lambda_c + J_{br}^\top(q_r)\,\lambda_t, \tag{2.1}$$

where $D_r(q_r) \in \mathbb{R}^{18\times 18}$ is the symmetric and positive definite mass-inertia matrix, $H_r(q_r, \dot{q}_r) \in \mathbb{R}^{18}$ denotes the Coriolis, centrifugal, and gravitational terms, and $B_r \in \mathbb{R}^{18\times 12}$ is the input distribution matrix. In addition, $J_c(q_r) \in \mathbb{R}^{3n_c\times 18}$ is the Jacobian at the contacting foot ends, and $J_{br}(q_r) := \frac{\partial p_{br}}{\partial q_r}(q_r) \in \mathbb{R}^{6\times 18}$ is the Jacobian of the point on the robot to which the tail is rigidly attached.

The tail is modeled as an 8 DOF system where the first 6 DOF represent a floating base,

and the last 2 DOF represent the internal roll and yaw of the tail. The model uses a floating base in order to make it modular with other systems, where the interacting dynamics may be coupled with Lagrange multipliers. Furthermore, in creating the dynamics with a floating base, that also allows for a great amount of flexibility in choosing where the tail is positioned on the robot without needing to re-derive the equations of motion of the quadruped or the tail.

The tail is modeled using DH-convention and all non-conservative forces are neglected. Using MATLAB the transformation matrices from one link to the next were symbolically calculated, where the transformation between yaw links included the same symbolic rotation as the previous, therefore enforcing the holonomic constraints. The transformations between the joints and COMs and their subsequent linear and angular Jacobians were obtained in order to use the Lagrange method to derive the dynamics of the system. However, it is important to note that if the number of links exceeded 8 then the dynamics proved to be too complex for MATLAB to handle symbolically. In order to alleviate this problem, an interface to the Mathematica symbolic engine is used from MATLAB to solve more complex symbolic equations. This allows for continuity as all code is written in C++ and MATLAB. Once derived, the dynamics were exported to mex files and C++ files which are used for simulation and control. The resulting dynamics were numerically validated using the Rigid Body Dynamics Library (RBDL) [132] which uses a recursive Newton-Euler method, and the dynamics matched to approximately machine precision.

The generalized coordinates (see Fig. 2.1) of the tail are defined as $q_t := \text{col}(p_t, \alpha_t, \theta) \in \mathcal{Q}_t \subset \mathbb{R}^8$, where the subscript "$t$" stands for the tail, $p_t \in \mathbb{R}^3$ and $\alpha_t \in \mathbb{R}^3$ represent the absolute position and orientation of the base of the tail in the inertial world frame, respectively, and $\theta := \text{col}(\theta_{\text{roll}}, \theta_{\text{yaw}}) \in \mathbb{R}^2$ are the internal roll and yaw angles. Lastly, $\tau_t := \text{col}(\tau_{\text{roll}}, \tau_{\text{yaw}}) \in \mathbb{R}^2$ are the torque inputs to the tail roll and yaw actuators. The equations of motion can be

described as follows:

$$D_t(q_t)\,\ddot{q}_t + H_t(q_t, \dot{q}_t) = B_t\,\tau_t - J_{bt}^{\top}(q_t)\,\lambda_t \tag{2.2}$$

where $D_t(q_t) \in \mathbb{R}^{8\times 8}$, $H_t(q_t, \dot{q}_t) \in \mathbb{R}^8$, and $B_t \in \mathbb{R}^{8\times 2}$ denote the mass-inertia matrix, Coriolis, centrifugal, and gravitational forces, and input distribution matrix, respectively. Furthermore, $J_{bt}(q_t) := \frac{\partial p_{bt}}{\partial q_t}(q_t) \in \mathbb{R}^{6\times 8}$ is the Jacobian of the base of the tail in the tail inertial frame. The holonomic constraints between the robot and tail can be expressed as $p_{br}(q_r) = p_{bt}(q_t)$, in which $p_{br}(q_r) \in \mathbb{R}^6$ and $p_{bt}(q_t) \in \mathbb{R}^6$ represent the absolute position and orientation of the base of the tail in q_r and q_t coordinates, respectively. Differentiating the holonomic constraints twice yields

$$J_{br}(q_r)\,\ddot{q}_r + \dot{J}_{br}\,(q_r, \dot{q}_r) = J_{bt}(q_t)\,\ddot{q}_t + \dot{J}_{bt}\,(q_t, \dot{q}_t)\,. \tag{2.3}$$

Equations (2.1)-(2.3) may then be put into a system of algebraic differential equations to solve for the acceleration of each body and the interacting forces between them, that is,

$$\begin{bmatrix} D_r & \mathbf{0} & -J_{br}^{\top} \\ \mathbf{0} & D_t & J_{bt}^{\top} \\ J_{br} & -J_{bt} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{q}_r \\ \ddot{q}_t \\ \lambda_t \end{bmatrix} = \begin{bmatrix} B_r\,\tau_r - H_r + J_c^{\top}\lambda_c \\ B_t\,\tau_t - H_t \\ \dot{J}_{bt} - \dot{J}_{br} \end{bmatrix}, \tag{2.4}$$

where $\mathbf{0}$ represents a matrix of zeros of the correct dimension. Note that the final equation in (2.4) ensures that there is a rigid connection between the tail and the quadruped. The equations of motion in the nonlinear coupled dynamics (2.4) can be expressed as an input-affine system as follows:

$$\dot{z} = f(z) + g(z)\,\tau + w(z)\,\lambda_c \tag{2.5}$$

where $z := \mathrm{col}(q_r, q_t, \dot{q}_r, \dot{q}_t)$ denotes the state vector and $\tau := \mathrm{col}(\tau_r, \tau_t)$ represents the control inputs. We remark that the model (2.5) is valid if the GRFs at the contacting leg ends are

feasible. In other words, the GRFs must take values in the friction cone defined as follows:

$$\mathcal{FC} := \left\{ \text{col}(F_x, F_y, F_z) \,|\, F_z > 0,\, \pm F_x < \frac{\mu}{\sqrt{2}} F_z,\, \pm F_y < \frac{\mu}{\sqrt{2}} F_z \right\}, \tag{2.6}$$

where μ is the friction coefficient.

Chapter 3

Quasi-Static Quadrupedal Locomotion with Robotic Tails

The objective of this chapter is to provide an overview of the standard LIP model, followed by the derivation of the the extended LIP model wherein the tail has been integrated. This chapter proceeds to develop both the high- and low-level control algorithms for planning and tracking, respectively, followed by a series of extensive simulations displaying the efficacy of the proposed approach. In particular, the simulations focus primarily on the robustness gained against external disturbances due to the addition of the tail.

3.1 Extended LIP Model

The purpose of this section is to derive a reduced-order model based on the LIP dynamics to study the equations of motion for the COM and the tail dynamics. The model, referred to as the extended LIP dynamics, will be utilized for the development of the real-time motion planning algorithm in Section 3.4.

The reduced-order LIP model can be described by [90]

$$\begin{bmatrix} \ddot{x}^{\text{com}} \\ \ddot{y}^{\text{com}} \end{bmatrix} = \frac{g_0}{z^{\text{com}}} \begin{bmatrix} x^{\text{com}} - x^{\text{cop}} \\ y^{\text{com}} - y^{\text{cop}} \end{bmatrix}, \tag{3.1}$$

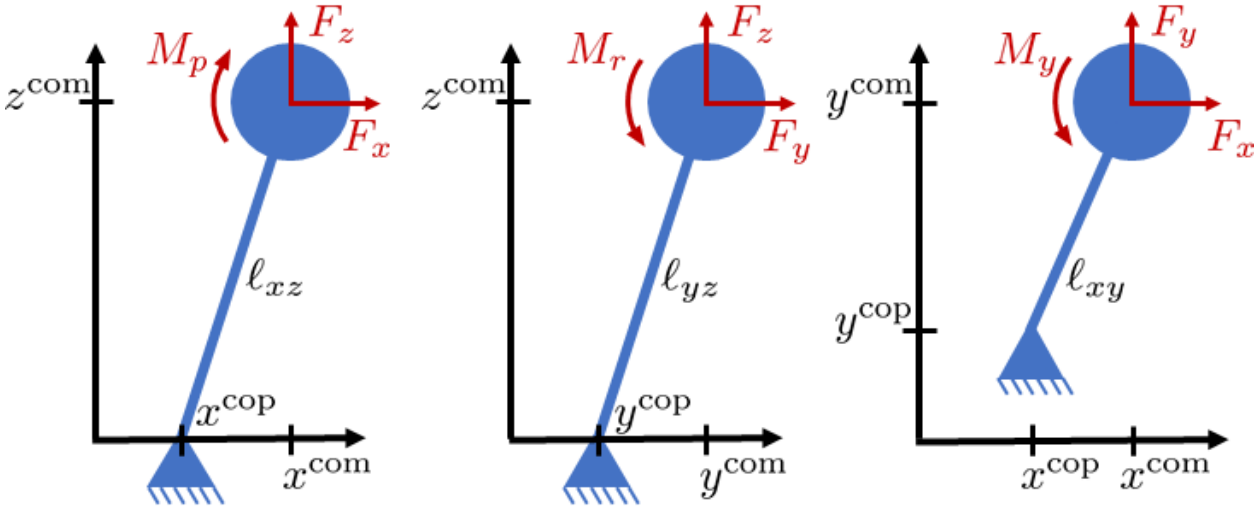

Figure 3.1: Free body diagram of the forces and moments induced by the tail on the LIP dynamics. Note that the z^{cop} is always zero.

where x^{com} and y^{com} represent the Cartesian coordinates of the COM with respect to the inertial world frame in the x and y directions, respectively, having been projected onto the xy-plane. In addition, z^{com} is the vertical height of the center of mass, which is assumed to remain constant throughout the duration of the gait, g_0 is the gravitational constant, and $u^{\text{cop}} := \text{col}(x^{\text{cop}}, y^{\text{cop}}) \in \mathbb{R}^2$ represents the Cartesian coordinates of the COP in the world frame. In order to include external forces in the LIP model, we consider the following dynamics

$$\begin{bmatrix} \ddot{x}^{\text{com}} \\ \ddot{y}^{\text{com}} \end{bmatrix} = \frac{1}{m_{\text{tot}}} \begin{bmatrix} F_x + F_{gx} + F_{M_p x} + F_{M_y x} \\ F_y + F_{gy} + F_{M_r y} + F_{M_y y} \end{bmatrix}, \tag{3.2}$$

where m_{tot} is the total robot mass, F_x denotes the external force in the x direction applied by the tail, F_{gx} represents the GRF in the x direction, $F_{M_r x}$ denotes the equivalent force in the x direction due to an external roll moment, $F_{M_p x}$ represents the equivalent force in the x direction due to an external pitch moment, and $F_{M_y x}$ denotes the equivalent force in the x direction due to an external yaw moment (see Fig. 3.1). The notation is defined similarly for both the y and z directions. In order to solve for the net GRF, we consider the equation

of motion in the z direction as

$$m_{\text{tot}}\,\ddot{z}^{\text{com}} = F_z + F_{gz} + F_{M_r z} + F_{M_p z} - m_{\text{tot}}\,g_0 \tag{3.3}$$

and let $\ddot{z}^{\text{com}} = 0$ for the duration of the gait. For this purpose, we first study the components of the GRF and the equivalent forces due to moments according to the geometry of the model. In particular, it can be shown that $F_{gx} = F_g \frac{x^{\text{com}} - x^{\text{cop}}}{l}$, $F_{gy} = F_g \frac{y^{\text{com}} - y^{\text{cop}}}{l}$, and $F_{gz} = F_g \frac{z^{\text{com}}}{l}$, where F_g denotes the net GRF and $l := \sqrt{(x^{\text{com}} - x^{\text{cop}})^2 + (y^{\text{com}} - y^{\text{cop}})^2 + (z^{\text{com}})^2}$. In addition, we can easily translate the moments into equivalent forces as follows (see Fig. 3.1)

$$\begin{aligned}
F_{M_p x} &= \frac{M_p}{\ell_{xz}^2} z^{\text{com}}, & F_{M_p z} &= \frac{-M_p}{\ell_{xz}^2} \left(x^{\text{com}} - x^{\text{cop}}\right) \\
F_{M_r y} &= \frac{-M_r}{\ell_{yz}^2} z^{\text{com}}, & F_{M_r z} &= \frac{M_r}{\ell_{yz}^2} \left(y^{\text{com}} - y^{\text{cop}}\right) \\
F_{M_y x} &= \frac{-M_y}{\ell_{xy}^2} \left(y^{\text{com}} - y^{\text{cop}}\right), & F_{M_y y} &= \frac{M_y}{\ell_{xy}^2} \left(x^{\text{com}} - x^{\text{cop}}\right),
\end{aligned}$$

where $\ell_{xz} := \sqrt{(x^{\text{com}} - x^{\text{cop}})^2 + (z^{\text{com}})^2}$ represents the distance between the COM and COP in the xz-plane, and in a similar manner, $\ell_{yz} := \sqrt{(y^{\text{com}} - y^{\text{cop}})^2 + (z^{\text{com}})^2}$ and $\ell_{xy} := \sqrt{(x^{\text{com}} - x^{\text{cop}})^2 + (y^{\text{com}} - y^{\text{cop}})^2}$. These geometric relations together with (3.3) and $\ddot{z}^{\text{com}} = 0$ finally result in the following net GRF

$$F_g = \frac{l}{z^{\text{com}}} \left(m_{\text{tot}} g_0 - F_z + \frac{M_p (x^{\text{com}} - x^{\text{cop}})}{(x^{\text{com}} - x^{\text{cop}})^2 + (z^{\text{com}})^2} - \frac{M_r (y^{\text{com}} - y^{\text{cop}})}{(y^{\text{com}} - y^{\text{cop}})^2 + (z^{\text{com}})^2} \right). \tag{3.4}$$

The GRF in (3.4) can then be combined with (3.2) to obtain the extended LIP dynamics which are nonlinear due to the existence of several inputs, including F_z and the moments (i.e., M_r, M_p, and M_y). However, the magnitude of these terms compared to the COP, F_x, and F_y inputs are assumed to be small. Furthermore, ℓ_{xy} goes to zero as the COM approaches the COP which causes $F_{M_y x}$ and $F_{M_y y}$ to become undefined due to division by

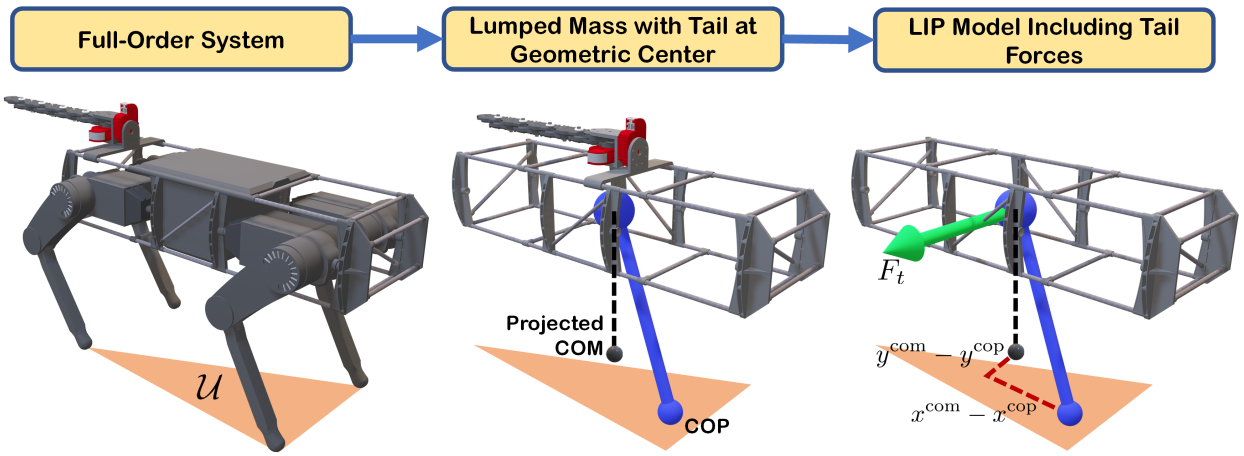

Figure 3.2: Illustration of the reduced-order extended LIP model.

zero. For these reasons we make the assumption that F_z and the moments are zero for the reduced-order model and treat these terms as a disturbance. We therefore use only the standard COP inputs (i.e., x^{cop} and y^{cop}) together with F_x and F_y which are to be applied by the tail. We can show that this assumption will reduce the nonlinear dynamics of the extended LIP model to the following linear dynamics

$$\begin{bmatrix} \ddot{x}^{\text{com}} \\ \ddot{y}^{\text{com}} \end{bmatrix} = \frac{g_0}{z^{\text{com}}} \begin{bmatrix} x^{\text{com}} - x^{\text{cop}} \\ y^{\text{com}} - y^{\text{cop}} \end{bmatrix} + \frac{1}{m_{\text{tot}}} \begin{bmatrix} F_x \\ F_y \end{bmatrix}. \tag{3.5}$$

3.2 Augmented LIP Dynamics

In order to properly plan for the movement of the tail in the high-level MPC, the nonlinear tail dynamics must be integrated with (3.5). For the sake of keeping the reduced-order model as simple as possible while capturing the overall dynamics, the tail is assumed to be attached to the geometric center of the robot, as shown in Fig. 3.2. Due to the nature of B_t and J_b,

the equations of motion in (2.2) can be rewritten more concisely as

$$D_t(q_t)\,\ddot{q}_t + H_t\,(q_t, \dot{q}_t) = \begin{bmatrix} -F_t \\ -M_t \\ \tau_t \end{bmatrix}, \tag{3.6}$$

where $\lambda_t = \mathrm{col}(F_t, M_t)$ with $F_t := \mathrm{col}(F_x, F_y, F_z) \in \mathbb{R}^3$ being the forces at the base of the tail in the x, y, and z directions, and $M_t := \mathrm{col}(M_r, M_p, M_y) \in \mathbb{R}^3$ being the moments at the base of the tail in the roll, pitch, and yaw directions.

Considering that we are only interested in the forces in the x and y directions, only the first two rows of (3.6) are used for the high-level MPC. We may then solve for the reaction forces caused by the tail in the x and y directions by noting the tail mass matrix, D_t, and nonlinear vector, H_t, may be decomposed into individual parts, denoted by D_{ij} for $1 \leq i, j \leq 8$ and H_i for $1 \leq i \leq 8$, respectively. More specifically, we have

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} -D_{11}\,\ddot{x}^{\mathrm{com}} - D_{17}\,\ddot{\theta}_{\mathrm{roll}} - D_{18}\,\ddot{\theta}_{\mathrm{yaw}} - H_1 \\ -D_{22}\,\ddot{y}^{\mathrm{com}} - D_{27}\,\ddot{\theta}_{\mathrm{roll}} - D_{28}\,\ddot{\theta}_{\mathrm{yaw}} - H_2 \end{bmatrix}, \tag{3.7}$$

where θ_{roll} and θ_{yaw} denote the actuated roll and yaw angles (i.e., shape variables) of the tail. Here, we assume that the roll, pitch, and yaw of the quadruped (and the base of the tail) are zero, and z^{com} is constant. Hence, (3.7) does not contain the portions of the mass-inertia matrix pertaining to the orientation and height of the base of the tail. We further remark that $D_{12} = D_{21} = 0$. Due to the nature of the floating base coordinates, the upper-left 3×3 block of the D matrix is diagonalized. Combining (3.5) and (3.7), the LIP equations with

external forces in the x and y directions may then be written as

$$\begin{bmatrix} \ddot{x}^{\text{com}} \\ \ddot{y}^{\text{com}} \end{bmatrix} = \begin{bmatrix} \frac{g_0}{z^{\text{com}}}(x^{\text{com}} - x^{\text{cop}}) + \frac{-D_{11}\ddot{x}^{\text{com}} - D_{17}\ddot{\theta}_{\text{roll}} - D_{18}\ddot{\theta}_{\text{yaw}} - H_1}{m_{\text{tot}}} \\ \frac{g_0}{z^{\text{com}}}(y^{\text{com}} - y^{\text{cop}}) + \frac{-D_{22}\ddot{y}^{\text{com}} - D_{27}\ddot{\theta}_{\text{roll}} - D_{28}\ddot{\theta}_{\text{yaw}} - H_2}{m_{\text{tot}}} \end{bmatrix}. \tag{3.8}$$

Similarly, the dynamics for the shape of the tail (i.e. $\ddot{\theta}_{\text{roll}}$ and $\ddot{\theta}_{\text{yaw}}$) may be written as

$$\begin{bmatrix} D_{71}\ddot{x}^{\text{com}} + D_{72}\ddot{y}^{\text{com}} + D_{77}\ddot{\theta}_{\text{roll}} + D_{78}\ddot{\theta}_{\text{yaw}} \\ D_{81}\ddot{x}^{\text{com}} + D_{82}\ddot{y}^{\text{com}} + D_{87}\ddot{\theta}_{\text{roll}} + D_{88}\ddot{\theta}_{\text{yaw}} \end{bmatrix} = \begin{bmatrix} \tau_{\text{roll}} - H_7 \\ \tau_{\text{yaw}} - H_8 \end{bmatrix}. \tag{3.9}$$

Combining (3.8) and (3.9), and moving all acceleration terms to the left hand side, we obtain the final equations for the reduced-order model as a set of algebraic differential equations of the form

$$W(x)\,\dot{x} = A\,x + B\,u + N(x), \tag{3.10}$$

where $x := \text{col}(x^{\text{com}}, y^{\text{com}}, \theta_{\text{roll}}, \theta_{\text{yaw}}, \dot{x}^{\text{com}}, \dot{y}^{\text{com}}, \dot{\theta}_{\text{roll}}, \dot{\theta}_{\text{yaw}}) \in \mathbb{R}^8$ and $u := \text{col}(x^{\text{cop}}, y^{\text{cop}}, \tau_{\text{roll}}, \tau_{\text{yaw}}) \in \mathbb{R}^4$ denote the state and control inputs, respectively. In addition, $W(x) \in \mathbb{R}^{8\times 8}$ is a weighting matrix, $A \in \mathbb{R}^{8\times 8}$ represents the state matrix, $B \in \mathbb{R}^{4\times 8}$ denotes the input distribution matrix, and $N(x) \in \mathbb{R}^8$ contains the remaining nonlinear terms.

3.3 Linearization and Discretization

The final reduced-order equations (3.10) are linearized by taking the first order Taylor series expansion. Due to the complexity of symbolically inverting the W matrix, the Taylor series

expansion is applied to both the left and right hand sides of the equation separately yielding

$$\begin{aligned} G(x,\dot{x}) &:= W(x)\,\dot{x} \\ &\approx G(x_0,\dot{x_0}) + \left.\frac{\partial G}{\partial x}\right|_{(x_0,\dot{x}_0)} (x-x_0) + \left.\frac{\partial G}{\partial \dot{x}}\right|_{(x_0,\dot{x}_0)} (\dot{x}-\dot{x}_0) \end{aligned} \tag{3.11}$$

and

$$\begin{aligned} F(x,u) &:= A\,x + B\,u + N(x) \\ &\approx F(x_0,u_0) + \left.\frac{\partial F}{\partial x}\right|_{(x_0,u_0)} (x-x_0) + \left.\frac{\partial F}{\partial u}\right|_{(x_0,u_0)} (u-u_0), \end{aligned} \tag{3.12}$$

where x_0, $\dot{x}_0$, and u_0 are the operating points around which the system is linearized. Combining (3.11) and (3.12) and solving for $\dot{x}$, we obtain the linearized equations of motion

$$\begin{aligned} \dot{x} = &\left.\frac{\partial G}{\partial \dot{x}}\right|^{-1}_{(x_0,\dot{x}_0)} \left(\left.\frac{\partial F}{\partial x}\right|_{(x_0,u_0)} - \left.\frac{\partial G}{\partial x}\right|_{(x_0,\dot{x}_0)} \right) (x-x_0) \\ &+ \left.\frac{\partial G}{\partial \dot{x}}\right|^{-1}_{(x_0,\dot{x}_0)} \left.\frac{\partial F}{\partial u}\right|_{(x_0,u_0)} (u-u_0) + \dot{x}_0 \end{aligned} \tag{3.13}$$

as $G(x_0,\dot{x}_0) = F(x_0,u_0)$. For future reference, one can rewrite (3.13) in a compact form as the following control affine system

$$\dot{x} = A_\ell\,(x_0,\dot{x}_0)\; x + B_\ell(x_0,u_0)\,u + d_\ell\,(x_0,\dot{x}_0,u_0)\,, \tag{3.14}$$

where the subscript "ℓ" stand for the linearized dynamics.

The dynamics (3.14) are then discretized via an Euler formulation such that $x_{k+1} = x_k + \dot{x}_k T_s$ where $k \in \mathbb{Z}_{\geq 0}$ and T_s is the sampling time. Using (3.14), we are left with a set of difference equations of the form

$$x_{k+1} = A_d\,x_k + B_d\,u_k + d, \tag{3.15}$$

where $A_d \in \mathbb{R}^{8\times 8}$ and $B_d \in \mathbb{R}^{8\times 4}$ are the discrete state and input distribution matrices, respectively, and $d \in \mathbb{R}^8$ is a vector of constants. The aforementioned matrices and vector are constant for a given linearization/discretization, and are reevaluated in an event-based manner for the real-time path planning problem (see Section 3.4).

3.4 High-Level Path Planner

The objective of this section is to formulate a real-time optimal control problem for the extended reduced-order model to compute an optimal trajectory for the COM and tail motions subject to the COP remaining within the support polygon created by the contacting leg ends, feasible net GRF, and joint and actuation limits of the tail. In particular, we formulate a high-level MPC problem to be solved in an event-based manner (i.e., beginning of each continuous-time domain) to drive the state of the extended reduced-order system from an arbitrary initial state, x_i, to some final state, x_f, over a finite number of continuous time domains, $M \geq 1$. We consider a general locomotion pattern with a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with the vertices set $\mathcal{V}$ and the edges set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ (see Fig. 3.3). The vertices denote the continuous-time domains of locomotion including double-, triple-, and quadruple-contact phases, and edges represent the discrete-time transitions between the continuous-time domains. Each continuous-time domain is assumed to have $N_d \geq 1$ evenly temporally spaced grid points over which the trajectory is optimized. In order to indicate the continuous-time domain for every time sample, we define the index function as $\zeta : \mathbb{Z}_{\geq 0} \to \{1, 2, \cdots, M\}$ by $\zeta[k] := \lfloor \frac{k}{N_d} \rfloor + 1$ for all $0 \leq k < M\,N_d$ and $\zeta[k] := M$ for $k \geq M\,N_d$. In our notation, $\lfloor . \rfloor$ represents the floor function.

MPC Formulation: We formulate an event-based optimal control problem for the real-time planning of the extended reduced-order model. In particular, we first linearize the

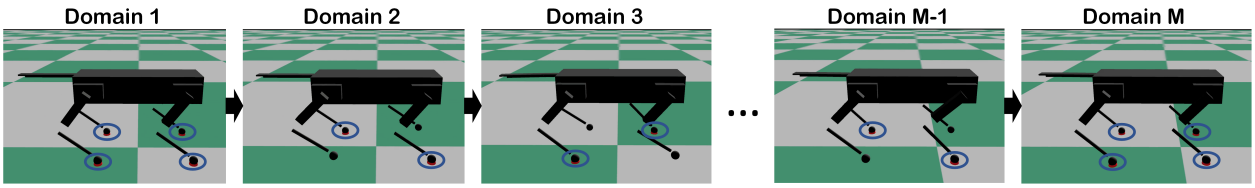

Figure 3.3: Snapshots of a nominal trot gait to illustrate the corresponding directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with M continuous-time domains. The first and last vertices of the graph represent quadruple-contact domains for starting and stopping, whereas the middle vertices represent double-contact domains. The contacting leg ends are depicted by the circles.

nonlinear extended LIP dynamics (3.10) about its current operating points at the event times (beginning of each continuous-time domain), that is, $k = r\,N_d$ for some $r \in \mathbb{Z}_{\geq 0}$. We then set up an MPC problem using the above-mentioned linearization over a finite control horizon $N_c = n\,N_d$ for some $n \geq 1$. This reduces the optimal control problem of the nonlinear system into QP that can be solved in real-time. The optimal control inputs are then employed for N_d grid points of the current domain while we discard the remaining control inputs for the next domains. At the beginning of the next domain (i.e., new event), the Jacobian linearization is updated according to the current state of the nonlinear reduced-order model and a new MPC problem is solved. This iterative procedure continues for all future events.

To make this notion more precise, let us assume that we are at the event time $k = r\,N_d$ for some $r \in \mathbb{Z}_{\geq 0}$. We next evaluate the Jacobian matrices A_ℓ and B_ℓ together with the vector d_ℓ as discussed in Section 3.3 for the following nonlinear dynamics

$$W(x_k^o)\,\dot{x}_k^o = F(x_k^o, u_{k-1}^o), \tag{3.16}$$

where x_k^o, $\dot{x}_k^o$, and u_{k-1}^o represent the current state of the original reduced-order system, the time derivative of the state, and the optimal control input for the previous time sample, respectively. Furthermore, the superscript "o" stands for the original system. These matrices are then kept constant for the formulation of the optimal control problem. At the next event,

the matrices will be reevaluated and will be utilized again for a new optimal control problem. We now consider the affine dynamics (3.15) with the initial condition $x_k = x_k^o$ for all future times. The optimal control inputs must be constrained such that u_k^{cop} (i.e., COP components of the input) belongs to the convex hull of the contacting points with the ground, $\mathcal{U}_{\zeta[k]} \subset \mathbb{R}^2$, for all domains. That is, $u_k^{\text{cop}} \in \mathcal{U}_{\zeta[k]}$, ensuring that the COP is within the support polygon for all time samples. Additionally, we constrain the net GRF to remain within the friction cone. From the linearized extended LIP dynamics, the friction cone constraints can be given by

$$\pm \left(A_f \, x_k + B_f \, u_k + d_f \right) \leq \gamma, \tag{3.17}$$

in which the subscript "f" stands for friction cone. More specifically, $A_f \in \mathbb{R}^{2\times 8}$ and $B_f \in \mathbb{R}^{2\times 4}$ are the state and input matrices which are composed of the rows of A_ℓ and B_ℓ pertaining to the accelerations in the x and y directions. In a similar manner, $d_f \in \mathbb{R}^2$ is composed of the appropriate rows of d_ℓ. Lastly, we define $\gamma := \mathbf{1} \frac{g_0 \mu}{\sqrt{2}}$ to represent the friction cone that the net GRF must stay within, where μ denotes the static friction coefficient, g_0 represents the gravitational constant, and $\mathbf{1}$ is an appropriately sized vector of ones.

Finally, because the tail joint positions and torques are explicitly included in the path planning problem, we may bound them in order to ensure shape and input feasibility. We define $\tau_{t,k} \in \mathbb{R}^2$ and $\theta_k \in \mathbb{R}^2$ to be the tail torque inputs and the actuated tail configurations at the time step k, respectively. We also define τ_t^{lb}, τ_t^{ub}, θ^{lb}, and θ^{ub} to denote the lower and upper bounds of the tail torque inputs and joint positions. We are now in a position to set up the MPC problem to be solved in an event-based manner (i.e., beginning of each continuous-time domain) in order to create an optimal sequence of inputs (i.e., COP and τ_t) which drives the system from x_0 to x_f over the control horizon. In particular, we consider the following

optimal control problem at $k = r\,N_d$

$$\min_{U_{k\to k+N_c-1|k}} \mathcal{J}_k\left(x_k, U_{k\to k+N_c-1|k}\right) = p\left(x_{k+N_c|k}\right) + \sum_{i=0}^{N_c-1} \mathcal{L}\left(x_{k+i|k}, u_{k+i|k}\right)$$

$$\begin{aligned}
\text{s.t.} \quad & x_{k+i+1|k} = A_d\, x_{k+i|k} + B_d\, u_{k+i|k} + d \\
& u^{\text{cop}}_{k+i|k} \in \mathcal{U}_{\zeta[k+i]} \\
& \pm\left(A_f\, x_{k+i|k} + B_f\, u_{k+i|k} + d_f\right) \leq \gamma \\
& \tau_t^{\text{lb}} \leq \tau_{t,k+i|k} \leq \tau_t^{\text{ub}} \\
& \theta^{\text{lb}} \leq \theta_{k+i|k} \leq \theta^{\text{ub}}, \quad i = 0, 1, \cdots, N_c - 1,
\end{aligned} \tag{3.18}$$

where $U_{k\to k+N_c-1|k} := \text{col}(u_{k|k}, \ldots, u_{k+N_c-1|k})$ denotes the sequence of control inputs and $x_{k+i|k}$ is the predicted state vector at time $k+i$ computed at time k subject to the dynamics $x_{k+i+1|k} = A_d\, x_{k+i|k} + B_d\, u_{k+i|k} + d$ with the initial condition $x_{k|k} := x_k$. In a similar manner, we denote $u_{k+i|k}$ to be the COP and tail torque inputs at time $k+i$ computed at time k. The terminal cost is expressed as $p(x_{k+N_c|k}) := ||x_{k+N_c|k} - x^{\text{des}}_{k+N_c|k}||^2_P$ for some symmetric positive definite matrix $P \in \mathbb{R}^{8\times 8}$. In this notation $x^{\text{des}}_{k+i|k}$ represents the desired state vector at time $k+i$ which is a smooth trajectory beginning at x_k and ending at x_f, and $||x||^2_P := x^\top P x$. The stage cost is also defined as $\mathcal{L}(x_{k+i|k}, u_{k+i|k}) := ||x_{k+i|k} - x^{\text{des}}_{k+i|k}||^2_Q + ||u_{k+i|k}||^2_R$, where $Q \in \mathbb{R}^{8\times 8}$ and $R \in \mathbb{R}^{4\times 4}$ are symmetric positive definite matrices.

Remark 3.1. The matrices A_d and B_d together with the vector d are reevaluated at the beginning of each new continuous-time domain using the current state of the original system and the previous optimal input, and are held constant for the duration of the MPC. The same is true for the friction matrices A_f and B_f, and the friction vector d_f. This reduces the MPC into QP that can be effectively solved in real-time (see Section 3.6 for more details).

Remark 3.2. The optimal COM and tail motions for the reduced-order model that are prescribed by the MPC for the current domain will be utilized as a reference trajectory

for the full-order model in Section 3.5. In particular, the low-level nonlinear controller will impose the full-order dynamics to asymptotically track the optimal reduced-order trajectory while imposing all contact forces to remain within the friction cone.

3.5 Low-Level Nonlinear Controller

This section presents the low-level controller (i.e., whole-body motion controller) which is developed based on virtual constraints and QP to track the optimal reduced-order motions generated by the high-level path planner. Specifically, we derive a nonlinear control algorithm which tracks the COM and tail trajectories prescribed by the event-based MPC for the current domain. Here, we extend the low-level nonlinear control algorithm of [66] to locomotion with robotic tails. While the path planner directly provides torques for the tail, the torques are based upon the linearized model and are not likely to be accurate enough to use on the full-order system. In addition to tracking the trajectories of the COM and the tail, the low-level controller is responsible for coordinating the motion of the swing legs such that they arrive at the next desired foot placement while ensuring the feasibility of all GRFs at the contacting legs.

We consider the following time-varying virtual constraints

$$y(z,t) := h(q,t) := h_0(q) - h_d(s,\alpha), \tag{3.19}$$

where $h_0(q)$ represents a set of holonomic quantities to be controlled, referred to as controlled variables, and $h_d(s,\alpha)$ denotes the desired evolution of the controlled variables h_0 on the gait. In particular, $h_d(s,\alpha)$ is taken as a Bézier polynomial that passes through the discrete waypoints generated by the MPC path planner, in which s is a gait phasing variable taken

as the normalized time and α represents the coefficients of the Bézier polynomial. The dimension of the outputs vary depending on the number of contact points with the ground. During quadruple-contact, h_0 consists of 8 components including the COM position, body orientation (i.e., roll, pitch, and yaw), and two internal joints of the tail. The idea here is to impose the full-order system to track the prescribed COM and tail motions while regulating the Euler angles of the floating base. During triple- and double-contact domains, these controlled variables are augmented with the Cartesian coordinates of the swing legs ends for the foot placement. Here, we would like the swing leg ends follow a desired trajectory in the work space starting from the former foothold and ending at the next one. This increases the dimension of the controlled variables to 11 and 14 during the triple- and double-contact domains, respectively.

Differentiating (3.19) along (2.5) we obtain the following output dynamics using I-O linearization

$$\begin{aligned}\ddot{y} &= \mathrm{L}_g\mathrm{L}_f y(z,t)\,\tau + \mathrm{L}_c\mathrm{L}_f y(z,t)\,\lambda_c + \mathrm{L}_f^2 y(z,t) - \frac{\partial^2 y}{\partial t^2}(z,t) \\ &= -K_D\,\dot{y} - K_P\,y, \end{aligned} \tag{3.20}$$

where $\mathrm{L}_g\mathrm{L}_f y$, $\mathrm{L}_c\mathrm{L}_f y$, and $\mathrm{L}_f^2 y$ are Lie derivatives and K_P and K_D are positive definite gain matrices. Furthermore, because the GRFs cannot be measured on the quadruped, we estimate the GRFs through the use of a rigid contact model. This assumes that the acceleration of the foot remains zero after contact which may be expressed as $\ddot{p} = J_c(q_r)\,\ddot{q}_r + \frac{\partial}{\partial q_r}\left(J_c(q_r)\,\dot{q}_r\right)\dot{q}_r = 0$. This in tandem with (2.5) yields

$$\ddot{p} = \mathrm{L}_g\mathrm{L}_f p(z)\,\tau + \mathrm{L}_c\mathrm{L}_f p(z)\,\lambda_c + \mathrm{L}_f^2 p(z) = 0. \tag{3.21}$$

Analogous to [66], a convex QP is set up to solve for the optimal joint-level torques τ for the

robot as well as the tail, while imposing the feasibility of GRFs at all contacting legs. More precisely, we consider the following real-time QP

$$
\begin{aligned}
\min_{(\tau,\lambda_c)} \quad & \frac{\Gamma_\tau}{2}\|\tau\|^2 + \frac{\Gamma_\delta}{2}\|\delta\|^2 \\
& \begin{bmatrix} \mathrm{L}_g\mathrm{L}_f y & \mathrm{L}_c\mathrm{L}_f y \\ \mathrm{L}_g\mathrm{L}_f p & \mathrm{L}_c\mathrm{L}_f p \end{bmatrix} \begin{bmatrix} \tau \\ \lambda_c \end{bmatrix} + \begin{bmatrix} \delta \\ 0 \end{bmatrix} = \begin{bmatrix} -K_D\,\dot{y} - K_P\,y - \mathrm{L}_f^2 y - \frac{\partial^2 y}{\partial t^2} \\ -\mathrm{L}_f^2 p \end{bmatrix} \\
& \lambda_c \in \mathcal{FC} \\
& |\delta| \leq \delta^{\mathrm{ub}} \\
& \tau^{\mathrm{lb}} \leq \tau \leq \tau^{\mathrm{ub}},
\end{aligned} \tag{3.22}
$$

where $\mathcal{FC}$ denotes the friction cone, and τ^{lb} and τ^{ub} represent the lower and upper bounds for the torque inputs, respectively. In order to account for scenarios in which the I-O linearization cannot be achieved perfectly, the defect variable δ is introduced. The second term in the cost function is added to ensure that the defect variable remains small, where the 2-norm of the defect variable is weighted by some positive definite constant, Γ_δ. Furthermore, an upper bound is placed on $|\delta|$, denoted by δ^{ub}. The low-level QP for nonlinear control is executed in real-time (e.g., 1kHz) to compute the optimal torques for the full-order model of locomotion, while the higher-level MPC is solved at a slower rate (i.e., in an event-based manner) for trajectory re-planing based on the extended LIP model.

3.6 Numerical Simulations

The objective of this section is to numerically verify the effectiveness of the proposed control algorithm for robust and stable quadrupedal locomotion with robotic tails. We present a multitude of full-order simulations to demonstrate the increased robustness that an articu-

lated tail provides using the proposed control technique. In all cases, we consider a trot gait with start and stop conditions as shown in Fig. 3.3.

3.6.1 Nominal Simulations

The extended LIP model is discretized using a sampling time of $T_s = 80$ (ms). The high-level MPC computes the optimal reduced-order trajectory over a control horizon of $n = 2$ domains, where each domain consists of $N_d = 4$ grid points. We utilize the optimal solution for the current domain, meaning that only the first four optimal trajectory points from the high-level MPC—which correspond to the current domain—are used as inputs to the low-level nonlinear controller. The rest of the parameters for the MPC are taken as $P = \text{diag}(10^4, 10^4, 10^3, 10^3, 10^4, 10^4, 1, 1)$, $Q = \text{diag}(10, 10, 10, 10, 0.1, 0.1, 0.1, 0.1)$, and $R = I_{4\times4}$, where I represents the identity matrix and "diag" denotes the diagonalization operator. Conceptually, we aim to give the MPC the ability to deviate from the desired tail trajectory (which is always 0) and prioritize tracking the desired COM motion. Additionally, we attempt to drive the tail back to zero at the end of the path planner to allow the tail to begin in an uncompromised position at the start of the next domain. However, we put relatively little weight on the velocity of the tail at the end of each domain, thereby allowing the tail to keep momentum. We begin by simulating the original and extended LIP models integrated with the proposed MPC approach to show the effectiveness of the higher-level controller in rejecting pulse-like disturbances, see Fig. 3.4. Here, we consider a forward trot gait with 20 domains subject to a pulse disturbance with a magnitude of 30 (N) in the lateral direction (i.e., y-axis) which is applied for the duration of two grid points, i.e., $2T_s = 160$ (ms), in domain 11. The reduced-order simulation is done in MATLAB and uses ECOS [133] in order to solve the QP arising from the event-based MPC. The stability of final target point for both the standard LIP (without tail) and the extended LIP (with tail) models subject

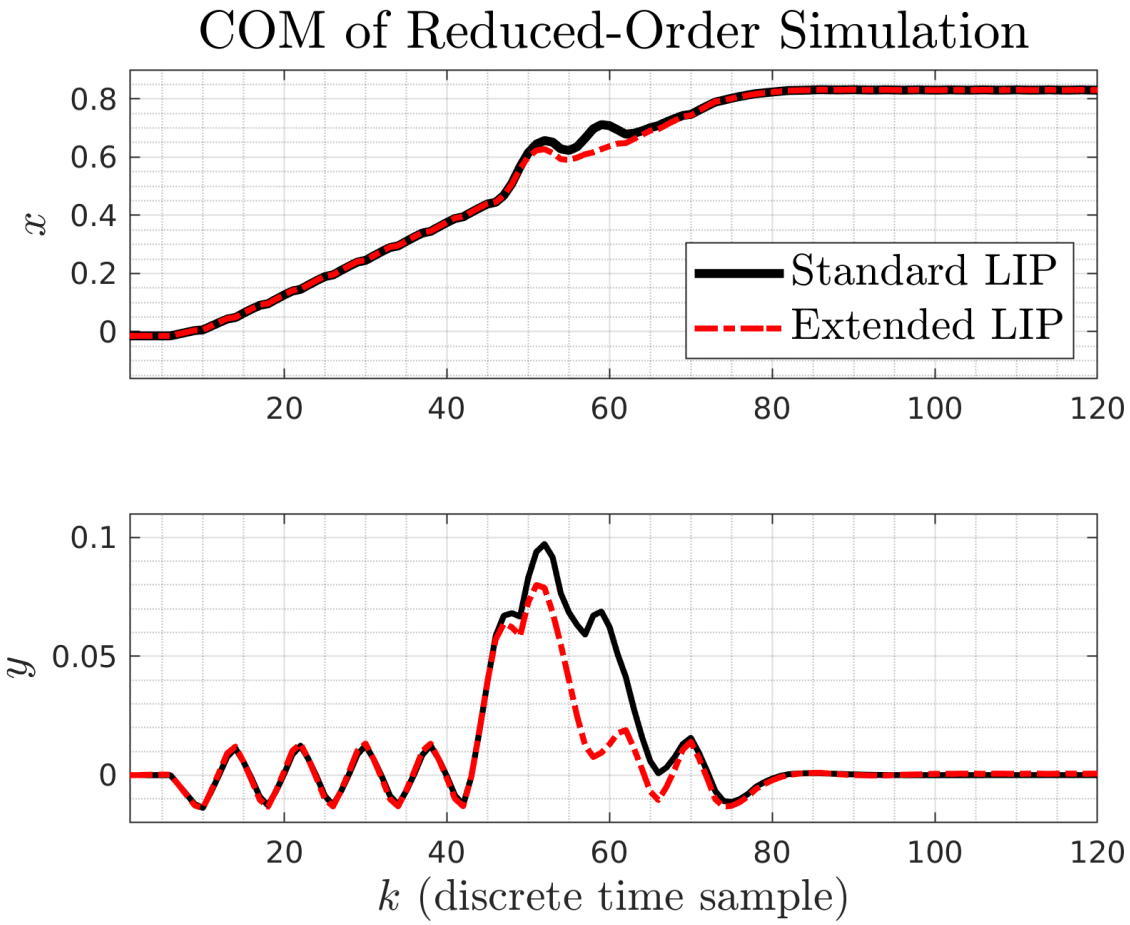

Figure 3.4: Reduced-order simulation of the event-based MPC acting on the standard LIP model (without tail) and the extended LIP model (with tail) for a forward trot with a 10 (cm) step length. Here we consider a 30 (N) pulse disturbance in the lateral direction which is applied for the duration of two MPC grid points (i.e., $2T_s$), starting at time sample 41.

to the event-based MPC is clear, but it is evident that the closed-loop system using the extended LIP model more adequately attenuates the disturbance. In particular, the system without the tail obtains a maximum displacement from the y-axis of approximately 9.7 (cm) while the system with the tail reaches a maximum displacement of 8 (cm) and settles back to steady state faster.

Full-order simulations are then done primarily in RaiSim [9], where we utilize qpSWIFT [134] to solve the high- and low-level QPs. Here, the MPC problem (3.18) is solved at the

beginning of each continuous-time domain, that is approximately every $4T_s = 320$ (ms), whereas the low-level nonlinear control QP is solved at 1kHz. We can show that the QP for the higher-level MPC has 120, 112, and 128 decision variables for the first, middle, and last continuous-time domains, respectively. In addition, the typical computation time of the MPC problem on a desktop computer with an Intel® Xeon® W-2125 CPU at 4.00 GHz (4 cores) and 16 GB of RAM is 0.32 (ms) to 0.42 (ms). The lower-level QP has 40 decision variables for both the double- and quadruple-contact domains of the trot gait with a typical computation time of 0.19 (ms) to 0.3 (ms).

RaiSim uses a universal robot description file (URDF) representation of the system. This presents a hurdle in that URDFs cannot represent holonomically constrained systems. In order to get around this, a URDF is created assuming that each yaw link in the tail is actuated instead of being holonomically constrained. In order to get the same dynamical behavior of the holonomically constrained tail in Fig. 2.1, RBDL [132] is then employed to compute the equivalent torque at each yaw joint of the tail via inverse dynamics by using τ_t and λ_t computed by the low-level QP.

We consider three nominal simulations, two of which are conducted in RaiSim, and the third in MATLAB. We consider multiple simulation environments in order to demonstrate the robustness of the closed-loop system against non-parametric uncertainties arising from different contact models. The robot hardware has a small compliant element at the toe, but the provided damping is assumed to be small and the rebound during impact is assumed to be zero. It is therefore reasonable to assume that the compliant dynamics at the toe are negligible when compared to the overall dynamics and impacts may be treated as rigid. For this reason, the majority of the simulations are conducted in RaiSim [9], where a rigid contact model between the leg ends and the ground is used, whereas few simulations are conducted in MATLAB, in which we utilize a LuGre model [135] to represent a compliant

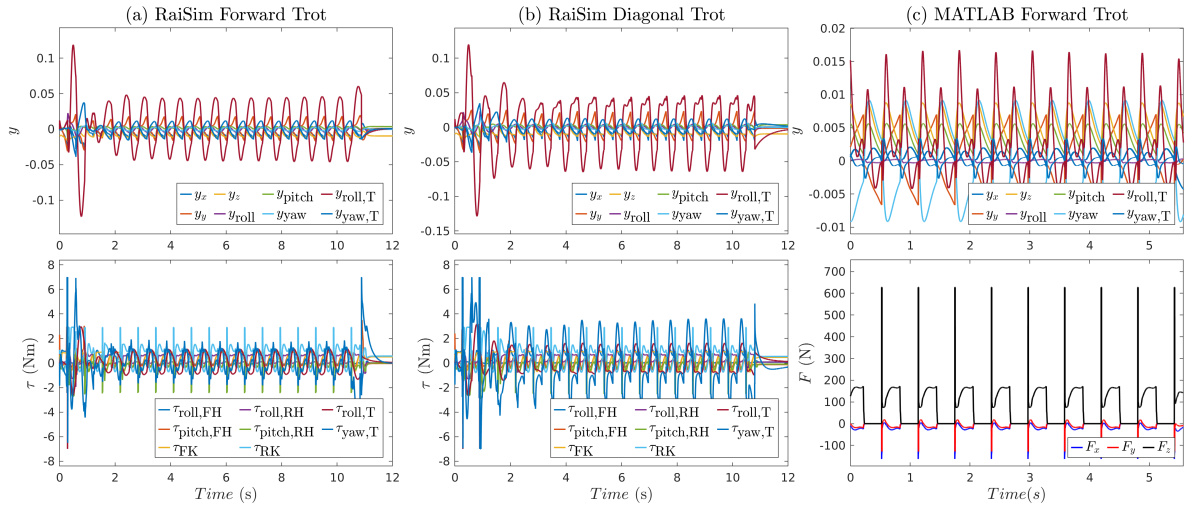

Figure 3.5: (a) and (b) Time profiles of the outputs (i.e., virtual constraints) and joint torques for the forward and diagonal trot gaits in RaiSim. (c) Time profile of the virtual constraints and the GRF experienced at one of the leg ends during forward trot gait with a LuGre contact model in MATLAB. Here, we only plot the components of the virtual constraints that correspond to the absolute position of the COM (i.e., y_x, y_y, and y_z), absolute orientation of the robot's floating base (i.e., y_{roll}, y_{pitch}, and y_{yaw}), and shape variables of the tail (i.e., $y_{\text{roll,T}}$ and $y_{\text{yaw,T}}$). Other components are not shown to simplify the illustration. For the torque plots, the subscripts "FH", "FK", "RH", "RK", and "T" stand for the front hip, front knee, rear hip, and rear knee of the left-hand-side of the robot as well as the tail, respectively. In addition, "roll" and "pitch" for the torque plots represent the roll and pitch motions of the joints.

contact model. Figures 3.5 (a) and (b) depict the output (i.e., virtual constraints) and torque profiles of the quadrupedal robot and tail for forward and diagonal trots in RaiSim. Figure 3.5 (c) shows a full-order simulation of the system run in MATLAB/Simulink subject to LuGre contact model. In addition, Fig. 3.5 (c) illustrates the components of the ground reaction force experienced at one of the legs. For the purpose of this paper, we use a step length of 10 (cm) for the forward trot and step lengths of (7,4) (cm) in the x and y directions for the diagonal trot.

3.6.2 Robustness Analysis

In order to study the robustness of the closed-loop system against external disturbances, we consider two different trot gaits, namely forward and diagonal trots. The robustness simulations fall under two different categories, the first being a pulse-like disturbance and the latter being a time-varying disturbance.

Pulse Disturbances

In order to analyze the closed-loop system's robustness, we investigate the effects of a pulse perturbation applied to the system for 200 (ms) at the COM. First we consider the forward trot gait with two different pulse disturbances, the first being 75 (N) in the lateral direction, and the second being 100 (N) in the forward direction. The virtual constraints and joint-level torques for these simulations are depicted in Figs. 3.6 (a) and (b). We also consider two additional simulations with the diagonal trot gait, where the first simulation is subject to a pulse disturbance directly opposing the direction of motion (i.e., along the unit vector $(-0.8682, -0.4961, 0)$), and the second is subject to a pulse disturbance which is orthogonal to the direction of motion (i.e., along the unit vector $(0.4961, -0.8682, 0)$). The pulse-like forces in the latter two simulations have magnitudes of 75 (N) and 110 (N), respectively, and the corresponding virtual constraints and torques are shown in Figs. 3.6 (c) and (d). The same disturbances were then applied to the system without the addition of the tail and the corresponding plots are provided in Fig. 3.7 to allow direct comparison. Figure 3.8 furthermore provides snapshots for the simulations found in Fig. 3.6 (a) and Fig. 3.7 (a). It is clear that the quadruped robot with the tail and the proposed control algorithm can successfully reach the final target in the presence of the aforementioned disturbance, whereas its counterpart without the tail cannot reach the target destination.

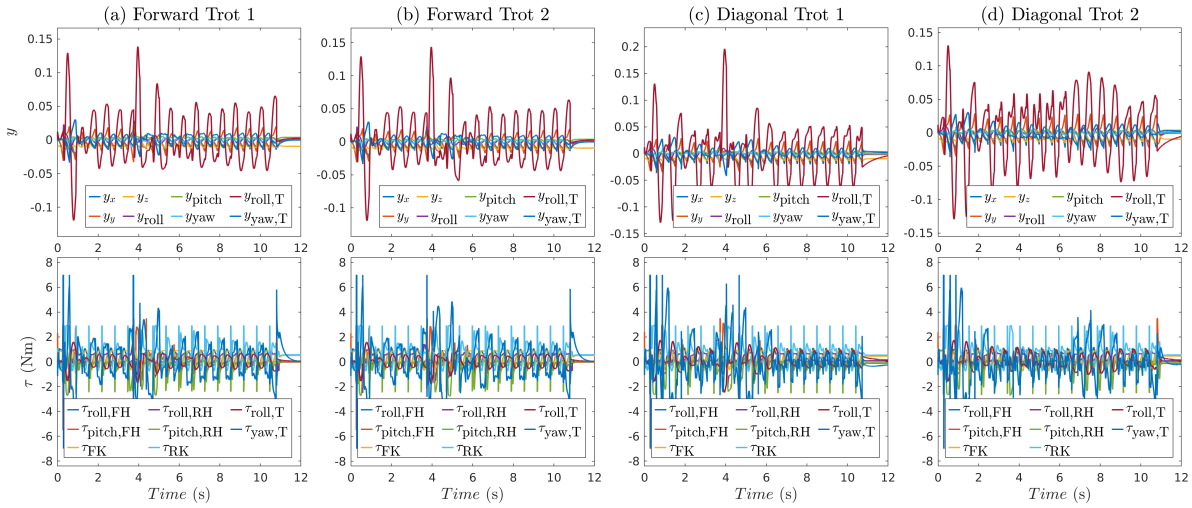

Figure 3.6: Time profiles of the outputs (i.e., virtual constraints) and joint torques for quadrupedal locomotion with the robotic tail subject to various pulse disturbances, all beginning at 3.5 (s) and lasting 200 (ms). The proposed controller can successfully reject the effect of disturbances while keeping the output and input profiles bounded. (a) Closed-loop system's behavior for a forward trot gait subject to a pulse disturbance of -75 (N) in the y direction. (b) Closed-loop system's behavior for a forward trot gait subject to a pulse disturbance of 100 (N) in the forward direction. (c) Closed-loop system's behavior for a diagonal trot gait subject to a pulse disturbance of 75 (N) orthogonal to the direction of locomotion in the xy-plane. (d) Closed-loop system's behavior for a diagonal trot gait subject to a pulse disturbance of 110 (N) opposing the direction of locomotion.

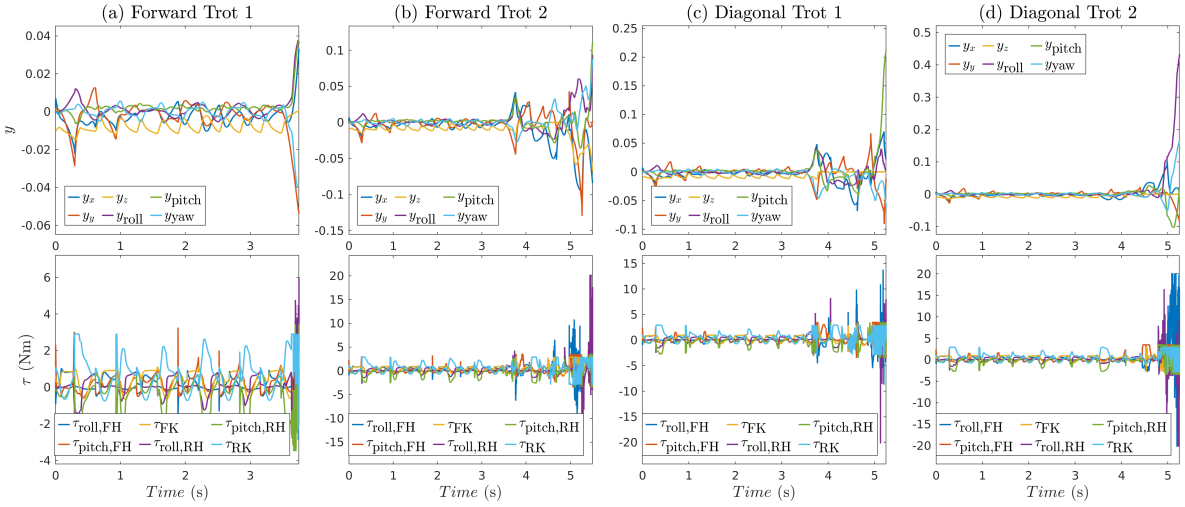

Figure 3.7: Time profiles of the outputs (i.e., virtual constraints) and joint torques for quadrupedal locomotion without the robotic tail subject to various pulse disturbances, all beginning at 3.5 (s) and lasting 200 (ms). Here, we use the same disturbances that are applied to the system with the robotic tail in Fig. 3.6. The instability of the gaits is clear. In particular, the outputs diverge after a while with the torques reaching the maximum values. (a) System's behavior for a forward trot gait subject to a pulse disturbance of -75 (N) in the y direction. (b) System's behavior for a forward trot gait subject to a pulse disturbance of 100 (N) in the forward direction. (c) System's behavior for a diagonal trot gait subject to a pulse disturbance of 75 (N) orthogonal to the direction of locomotion. (d) System's behavior for a diagonal trot gait subject to a pulse disturbance of 110 (N) opposing the direction of locomotion.

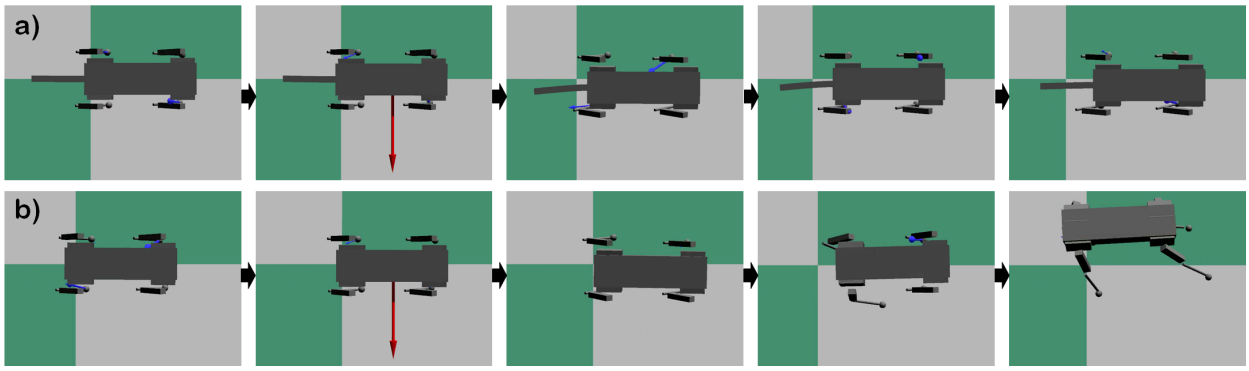

Figure 3.8: Snapshots of the quadrupedal locomotion with and without robotic tails and subject to the pulse-like disturbance in RaiSim. (a) Snapshots for the closed-loop system's behavior with the robotic tail as simulated in Fig. 3.6 (a). Here, we consider a forward trot gait subject to a pulse disturbance of -75 (N) in the y direction, starting at 3.5 (s) and lasting 200 (ms). (b) Snapshots for the system's behavior without the robotic tail and the same disturbance as simulated in Fig. 3.7 (a). The arrow illustrates the disturbance. The instability of the gait without the tail is clear.

It is important to note that the ability for the closed-loop system to reject a pulse disturbance is greatly influenced by when the disturbance is applied. Specifically, it depends on what point in the gait cycle the robot is currently in, and the state of the tail at that point. With this in mind, we aim to quantify the level of improvement the tail adds by considering a forward trot with a lateral pulse, where the pulse is applied at different times. In this regard, twelve different simulations were conducted, where in each simulation the pulse was applied 0.1 (s) later than the previous simulation. During each test, the largest magnitude pulse that the system was able to withstand while still reaching the target destination was recorded. On average, the system including the tail was able to sustain 61.7 (N) while still being able to reach the target destination which is an increase of 32.6% compared to its tailless counterpart. A similar procedure was done for the diagonal trot but was instead subject to a disturbance orthogonal to the direction of locomotion (i.e., along the unit vector $(0.4961, -0.8682, 0)$). The system with the tail could reject a pulse of 83.4 (N) on average. This is a 25.6% increase in magnitude compared to the system without the tail.

Time-Varying Persistent Disturbances

In this section, we consider four additional simulations in which the system is subject to a persistent time-varying disturbance. Similar to the pulse disturbances, we first investigate two forward trots, and then two diagonal trots, each subject to a different persistent disturbance. For each simulation, we note the largest magnitude at which the system can walk for 80 continuous-time domains both with and without the tail in order to compare the results. In the first scenario, we consider a forward trot gait with a lateral disturbance of the form $\beta \sin(4t)$, where the largest magnitude the system with the tail could withstand was $\beta = 21.5$ (N), which is 35.2% larger than the system without the tail. We then consider a second simulation of the forward trot gait subject to a disturbance of the form $\beta(\sin(4t), \cos(4t))$ in the x and y directions. In this case, the system with the tail could reject a maximum of $\beta = 11$ (N) while still being able to walk for 80 domains, up from $\beta = 7.4$ (N) without the tail. Figures 3.9 (a) and (b) depict the closed-loop system's behavior for the forward trot gait subject to the aforementioned time-varying disturbances. Finally, we consider two different simulations with a diagonal trot, where each system is subject to a disturbance of the form $\beta \sin(4t)$ but in different directions. During the first diagonal simulation, the disturbance acts along the direction of motion $\big($i.e., along the unit vector $(0.8682, 0.4961, 0)\big)$, and in the latter, the disturbance acts along the unit vector $(0.8682, -0.4961, 0)$. During these simulations, the system could reject $\beta = 14.3$ and $\beta = 18$ (N), which is an increase of 20.2% and 73% compared to the tailless system, respectively. Figures 3.9 (c) and (d) illustrate the time profiles of the virtual constraints and joint torques for these simulation. In order to demonstrate the robustness of the closed-loop system with the tail mechanism, Fig. 3.10 contains the same simulations having been performed on the system without the tail. In addition, Fig. 3.11 (a) and (b) provides snapshots of the simulations corresponding to Fig. 3.9 (b) and Fig. 3.10 (b), respectively.

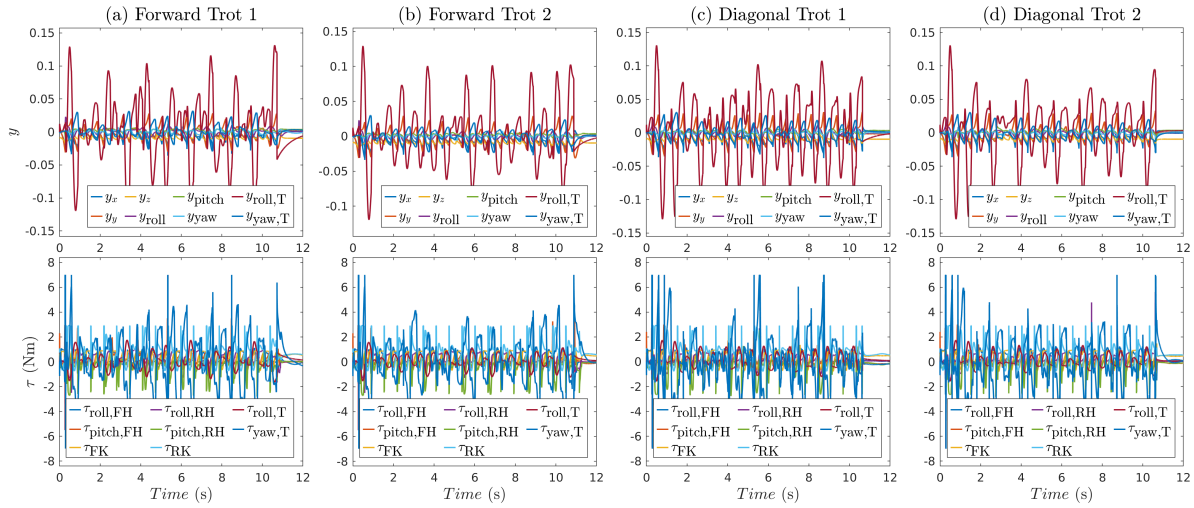

Figure 3.9: Time profiles of the outputs (i.e., virtual constraints) and joint torques for quadrupedal locomotion with the robotic tail subject to various sinusoidal disturbances. The proposed controller can successfully reject the effect of time-varying disturbances while keeping the output and input profiles bounded. (a) Closed-loop system's behavior for a forward trot gait subject to a disturbance of the form $18\sin(4t)$ (N) in the y direction. (b) Closed-loop system's behavior for a forward trot gait subject to a disturbance of the form $(8.5\sin(4t), 8.5\cos(4t))$ (N) in the x and y directions, respectively. (c) Closed-loop system's behavior for a diagonal trot gait subject to a disturbance of the form $13\sin(4t)$ (N) along the direction of locomotion. (d) Closed-loop system's behavior for a diagonal trot gait subject to a disturbance of the form $13\sin(4t)$ (N) along the unit vector $(0.8682, -0.4961, 0)$.

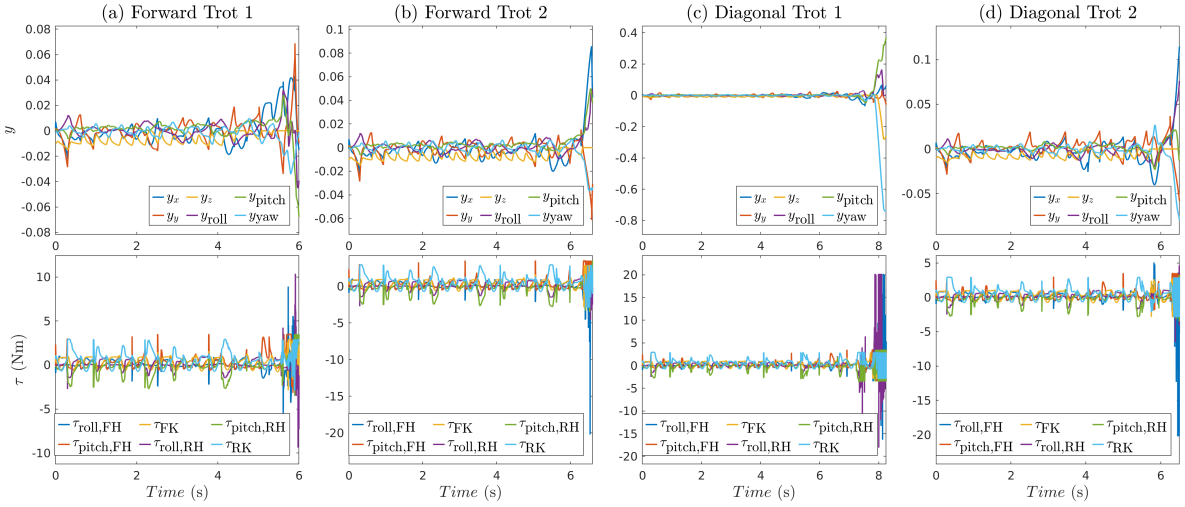

Figure 3.10: Time profiles of the outputs (i.e., virtual constraints) and joint torques for quadrupedal locomotion without the robotic tail subject to various sinusoidal disturbances. Here, we utilize the same time-varying disturbances that are applied to the system augmented with the robotic tail in Fig. 3.9. The instability of the gaits is clear. In particular, the outputs diverge after a while with the torques reaching the maximum values. (a) System's behavior for a forward trot gait subject to a disturbance of the form $18\sin(4t)$ (N) in the y direction. (b) System's behavior for a forward trot gait subject to a disturbance of the form $(8.5\sin(4t), 8.5\cos(4t))$ (N) in the x and y directions, respectively. (c) System's behavior for a diagonal trot gait subject to a disturbance of the form $13\sin(4t)$ (N) along the direction of locomotion. (d) System's behavior for a diagonal trot gait subject to a disturbance of the form $13\sin(4t)$ (N) along the unit vector $(0.8682, -0.4961, 0)$.

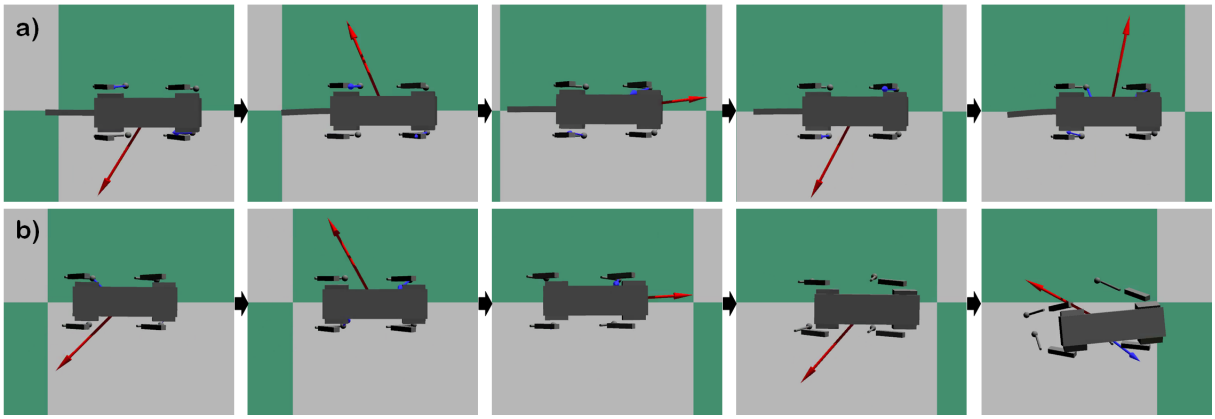

Figure 3.11: Snapshots of the quadrupedal locomotion with and without robotic tails and subject to a persistent sinusoidal disturbance in RaiSim. (a) Snapshots for the closed-loop system's behavior with the robotic tail as simulated in Fig. 3.9 (b). Here, we consider a forward trot gait subject to a time-varying disturbance of the form $(8.5\sin(4t), 8.5\cos(4t))$ (N) in the x and y directions. (b) Snapshots for the system's behavior without the robotic tail and the same disturbance as simulated in Fig. 3.10 (b). The arrow illustrates the disturbance. The instability of the gait without the tail is clear.

Chapter 4

Steps Toward Agile Quadrupedal Locomotion

While the preceding work demonstrates the increased robustness of locomotion when a quadruped is augmented with a tail, the results apply to quasi-static scenarios. In nature, biological organisms with tails have been observed to have increased stability largely in two scenarios, 1) during quasi-static or balance intensive tasks (i.e., traversing a balancing beam), and 2) during very dynamic motions (i.e., free falling with large initial angular momentum, high speed turning, etc.). Up to this point, only the first scenario has been examined. While this chapter does not consider the use of a tail for quadrupedal locomotion, the aim is to lay the groundwork to address robust and agile locomotion, particularly in a manner that is *suitable* to add a tail during future investigations.

In this chapter, we focus on the use of a more agile robot called A1[1], which is in a smaller class robot. It weighs approximately 12.45 (kg), and stands around 28 (cm) off the ground. Similar to V60 studied in the previous chapter, the A1 robot has 18 DOF including the absolute position and orientation of the COM, and hip roll, hip pitch, and knee pitch at each of the legs. The use of this robot is considered for all of the following work, including the simulations and experiments.

[1] https://unitree.com/

4.1 Foothold Selection

One of the limiting assumptions using the previous LIP-based controller is that the desired foothold locations are expected to be known in advance. Moreover, the locations are defined in the global reference frame. This is far from ideal in terms of robustness due to the fact that the desired footholds obtained at the beginning of the gait may not be optimal depending on certain environmental factors, including external disturbances, and uneven or shifting terrain. Here we consider the use of a very simple heuristic for choosing foothold locations in an online manner, which is often referred to as the Raibert Heuristic [136]. The target foothold location for each foot is prescribed relative to its corresponding hip, and is defined by [136]

$$X_{F,d} = \frac{\dot{x}^{\text{com}} T_{st}}{2} + K(\dot{x}^{\text{com}} - \dot{x}_d^{\text{com}}), \tag{4.1}$$

where $X_{F,d}$ is the desired foot displacement from the hip in the x direction, T_{st} is the stance time, K is a heuristic gain, and $\dot{x}_d^{\text{com}}$ is the desired COM velocity. The equations follow analogously for the foot step length in the y direction. Generally a new step length is determined in an event-based manner at impact, and is not updated during the swing phase. With that in mind, the COM and joint velocities undergo an abrupt change at the moment of impact, which is often referred to as a foot-strike transient. This is not ideal considering we are interested in maintaining an approximately constant *average* velocity. For this reason, (4.1) is modified slightly to use $\dot{\bar{x}}^{\text{com}}$ instead of $\dot{x}^{\text{com}}$, where $\dot{\bar{x}}^{\text{com}}$ is a moving average of the velocity to gain better insight into the trend of the velocity from the previous stance domain as opposed to using the instantaneous velocity, which is heavily influenced by impact. Finally, the step height is assumed to be zero for this work. Simulations of walking over rough terrain has shown this to be a reasonable assumption when combined with a

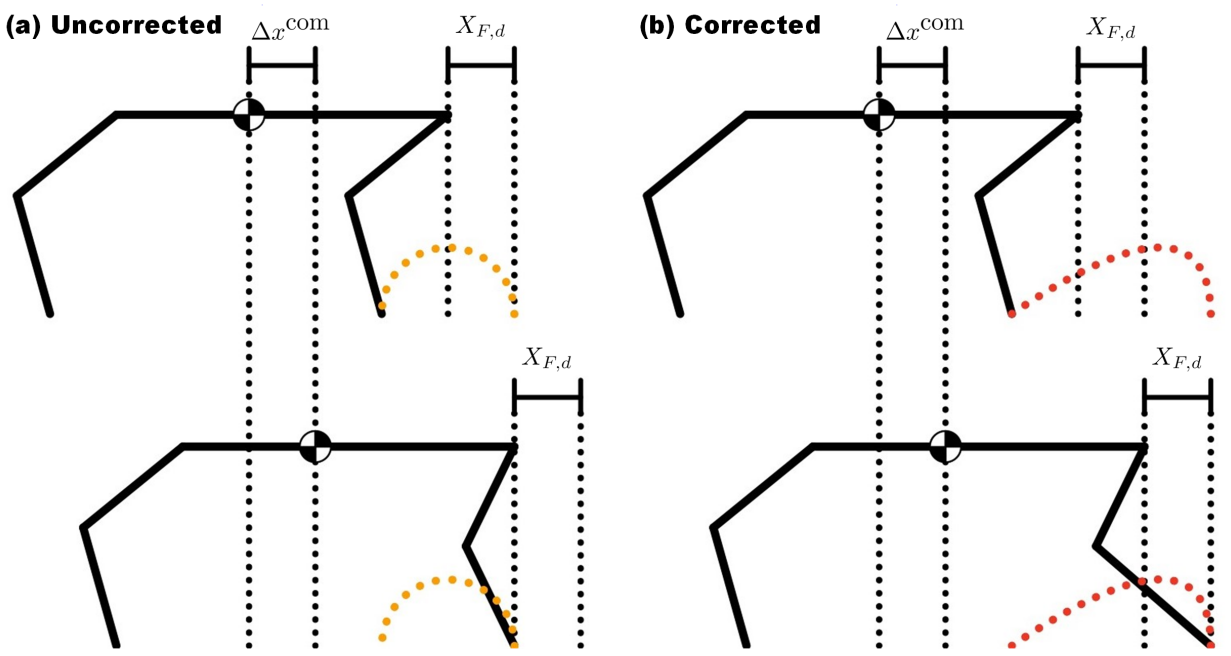

Figure 4.1: a) Depiction of the uncorrected foot placement strategy, in which the swing leg does not reach the desired step location relative to the hip since it does not consider the change in position of the COM as the domain evolves. b) the foot placement trajectory when using a time varying Bézier, wherein the desired relative step location is achieved. Note that in this particular case, the swing leg of the uncorrected trajectory ends up directly below the hip, as opposed to the correct distance of $X_{f,d}$ from the hip.

rudimentary early impact detection algorithm, but terrain height and orientation estimation will need to be added in the future to address hills, stairs, and other large sloping terrain.

The swing legs are set to track a Bézier polynomial which begins at the current swing leg position, and ends at the desired swing leg position as defined by (4.1). Considering that this foot placement method is relative to each respective hip, if the footstep location were prescribed in the global coordinates at the beginning of the domain, then we would expect poor tracking due to the fact that the body will be moving throughout the domain (see Fig. 4.1). One possible solution would be to add a correction term by simply setting the step

length according to

$$\begin{aligned} X_{F,d} &= 0.5\,\dot{\hat{x}}^{\text{com}} T_{st} + K(\dot{\hat{x}}^{\text{com}} - \dot{x}_d^{\text{com}}) + \dot{\hat{x}}^{\text{com}} T_{st} \\ &= 1.5\,\dot{\hat{x}}^{\text{com}} T_{st} + K(\dot{\hat{x}}^{\text{com}} - \dot{x}_d^{\text{com}}), \end{aligned}$$

but this assumption could easily be violated when in the presence of disturbances. To alleviate this issue, a time varying swing leg Bézier polynomial that considers the motion of the body and adjusts the polynomial accordingly throughout the swing domain is constructed. In particular, consider the standard Bézier polynomial $C(\alpha(t), s)$, where $\alpha(t)$ is a vector of Bézier node coefficients representing the trajectory that the swing leg is to follow, and $s = (t - t_0)/T$ is a normalized phasing variable with $T = t_f - t_0$. Here, t_f is the final anticipated time of the current domain, and t_0 is the initial time of the domain. Since the virtual constraints allow for the input of desired position, velocity, and acceleration, and since $\alpha(t)$ is a function of time, it can easily be verified that the rate of change of the Bézier curves are given by

$$\begin{aligned} C(\alpha, s) &= \sum_{i=0}^{n} b_{i,n}\,\alpha_i \quad \text{(Standard Bézier Curve)} \\ C_{TV}(\alpha(t), s) &= C(\alpha(t), s) \\ \dot{C}_{TV}(\alpha(t), \dot{\alpha}(t), s) &= \frac{\partial C}{\partial s}(\alpha(t), s)\frac{1}{T} + C(\dot{\alpha}(t), s) \\ \ddot{C}_{TV}(\alpha(t), \dot{\alpha}(t), \ddot{\alpha}(t), s) &= \frac{\partial^2 C}{\partial s^2}(\alpha(t), s)\frac{1}{T^2} + 2\frac{\partial C}{\partial s}(\dot{\alpha}(t), s)\frac{1}{T} + C(\ddot{\alpha}(t), s), \end{aligned}$$

where $b_{i,n}$ is a Bernstein basis polynomial of degree n, and TV denotes time varying. In this case, $\dot{\alpha}$ and $\ddot{\alpha}$ contain the velocity and acceleration of the respective hip in the global coordinate frame. Using these time-varying Bézier polynomials, the desired relative foot placement can be achieved much more accurately and robustly since the global foot placement

changes as the domain evolves.

This methodology for creating a swing foot trajectory, particularly the use of a time-varying Bézier polynomial, allows for other unique properties for safe locomotion when transferring to hardware. One major benefit of using Bézier polynomials is that they allow the user to define the tangent vector at the first an last nodes, meaning that we may enforce a desired swing leg velocity of zero for the beginning and end of the swing leg trajectory. This is especially important for the vertical direction to reduce impulsive forces placed on the motors and joints during foot strike. However, it is also of interest in the $x-$ and $y-$directions to help avoid foot slippage when foot contact occurs.

4.2 High-Level Path Planner

As previously mentioned, using the LIP model for the high-level path planner produces quasi-static gaits due to the necessity that the COP remains within the convex hull created by the contacting leg ends. Furthermore, while updating this model in an event-based manner does result in satisfactory locomotion and robustness, the event-based nature does not lend itself well to agile and high speed locomotion. To this end, producing fast and agile locomotion using LIP as the template model and solving in an event-based manner is unrealistic. For these reasons, we consider the use of a Single Rigid Body (SRB) model for the high-level path planner.

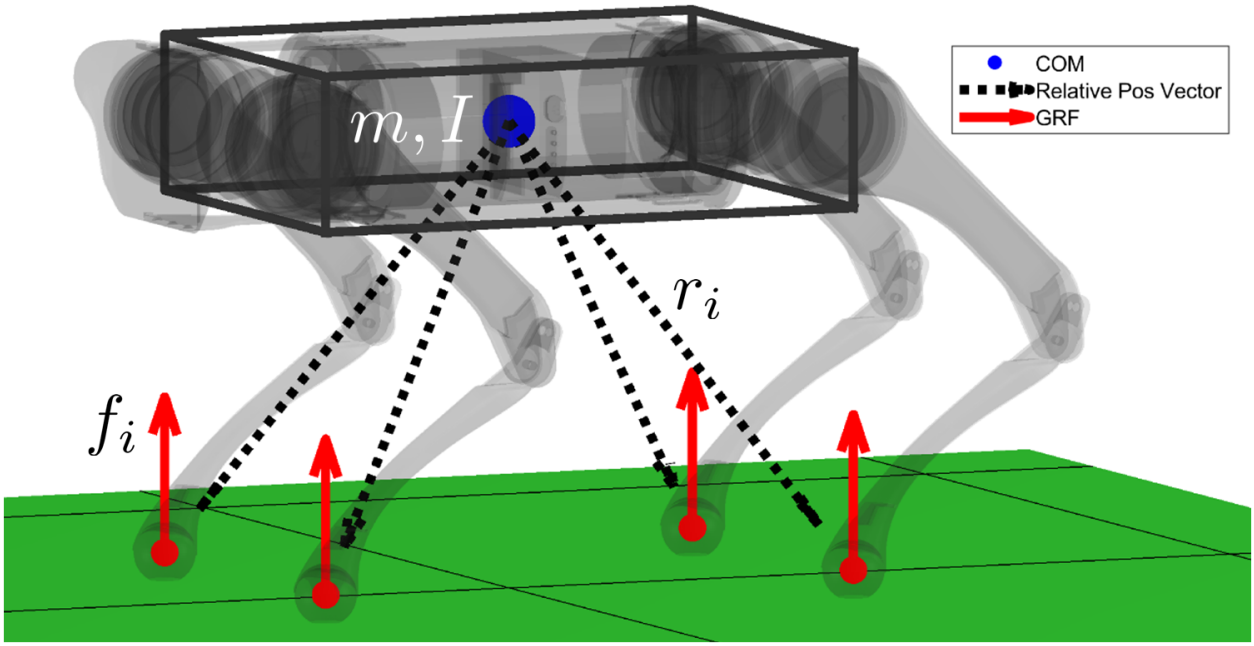

Figure 4.2: Illustration of the single rigid body model for quadrupedal locomotion. This model uses the relative vector from the COM to the feet, the forces applied by the contacting leg ends, and assumes all of the mass and inertia is concentrated in the torso, depicted here as a rectangle.

4.2.1 Single Rigid Body Dynamics

The SRB formulation for a quadruped is shown in Fig. 4.2 and is described by [92, 93, 94]

$$m\ddot{p}_r = f^{\text{net}} - mg_0$$
$$R^\top \tau^{\text{net}} = I\dot{\omega} + \hat{\omega} I\omega$$

where m is the total mass, g_0 is the gravitational constant, I is the body inertia (treated as constant), $p_r \in \mathbb{R}^3$ is the position of the COM of the robot in the inertial world frame, $\omega = \dot{\alpha}_r \in \mathbb{R}^3$ is the angular velocity in the body frame, $R \in^{3\times3}$ is the rotation matrix with respect to the inertia 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 on the COM. Furthermore, the forces

and torques are defined relative to the number of feet in contact with the ground and are calculated using

$$\begin{bmatrix} f^{\text{net}} \\ \tau^{\text{net}} \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} I_{3\times 3} \\ \hat{r}_i \end{bmatrix} f_i.$$

Here, f_i is the force applied by foot i, $\hat{r}_i$ is the distance between contacting foot i and the COM, and $\hat{(\cdot)}$ indicates a skew symmetric operator. The nonlinear reduced-order dynamics are then described by

$$\dot{x} = \frac{d}{dt} \begin{bmatrix} p \\ \dot{p} \\ R \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{p} \\ \frac{1}{m} f^{\text{net}} - g_0 \\ R\hat{\omega} \\ I^{-1}(R^\top \tau^{\text{net}} - \hat{\omega} I \omega) \end{bmatrix}. \tag{4.2}$$

It is important to note that this template model is setup in a manner which will allow a tail to be added in a relatively straight forward manner. In particular, the forces that the tail produces at its base can be added as an additional force on the rigid body. The primary challenge here is to fully define the range space of the forces that the tail can produce while in a given configuration. However, several previous works have focused on parameterizing the possible force production of various appendages via polytope approximations [137, 138, 139]. The concept could similarly be applied here, where the force polytope would then become a constraint in the MPC for the path planner.

4.2.2 Variational Based Linearization

There are two primary concerns when using (4.2), namely that it contains a matrix as one of the state variables, and that linearization could lead to singularities. We therefore opt to use Variational Based Linearization (VBL) and redefine our state in terms of error, which has successfully been implemented on other quadrupedal platforms (see e.g., [92, 93, 94]). The

primary goal of VBL is to linearize along the $SO(3)$ manifold, which the rotation evolves along, and parameterize the error in the Cartesian space via the exponential mapping. The variation w.r.t. the reference trajectory, R_d, is given by [140]

$$\delta R = \frac{\mathrm{d}}{\mathrm{d}\epsilon}\bigg|_{\epsilon=0} R_d \, \exp(\epsilon \, \hat{\eta}) = R_d \, \hat{\eta}$$

where $\eta \in \mathbb{R}^3$ is the approximate angle-axis error (describes the rotation needed to move from the current orientation to the desired orientation). The errors are then described by

$$\delta\omega = \hat{\omega} \, \eta + \dot{\eta} \tag{4.3}$$

$$e_R = \frac{1}{2} \left(R_d^\top R - R^\top R \right)^\vee \tag{4.4}$$

$$e_\omega = \omega - (R^\top R_d) \, \omega_d, \tag{4.5}$$

where $\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ maps a skew symmetric matrix to $\mathbb{R}^3$. The state of the linear reduced-order model is then given by

$$\sigma := \begin{bmatrix} \delta p_r \\ \delta \dot{p}_r \\ \eta \\ \delta\omega \end{bmatrix} \approx \begin{bmatrix} e_p \\ e_v \\ e_R \\ e_\omega \end{bmatrix} = \begin{bmatrix} p_r - p_d \\ \dot{p}_r - \dot{p}_d \\ \frac{1}{2} \left(R_d^\top R - R^\top R_d \right)^\vee \\ \omega - (R^\top R_d) \, \omega_d \end{bmatrix}, \tag{4.6}$$

where $(\cdot)_d$ indicates the desired variables, and $e_{(\cdot)}$ implies an error term. Analogous to the approach of [93], it can be shown that the linearized dynamics can be expressed as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\sigma = \begin{bmatrix} \delta\dot{p}_r \\ \frac{1}{m}\sum \delta f \\ -\hat{\omega}_d\eta + \delta\omega \\ I^{-1}\left[R_d^\top \widehat{\sum \tau_d^{\text{net}}}\eta + R_d^\top\left(\sum \hat{f}_d\delta p + \sum \hat{r}\delta f\right) + (\widehat{I\omega_d} - \hat{\omega}_d I)\delta\omega\right] \end{bmatrix}. \tag{4.7}$$

These reduced-order dynamics are then discretized via an Euler discretization and can be described compactly by

$$\sigma_{k+1} = A_d^{\text{VBL}}\,\sigma_k + B_d^{\text{VBL}}\,\delta f_k. \tag{4.8}$$

This methodology for parameterization of the linear system is slightly different than that used in [92, 94] in that, here, the error is described in terms of the angle axis error, whereas [92, 94] considers vectorization of the rotation matrix which results in an additional 6 state variables. However, it is worth noting that the linearization is similar in both works. The most notorious use of the SRB model for quadrupedal locomotion is the work involving the MIT Cheetah [91] (and its variants), though the linearization is accomplished via small angle approximations on the roll and pitch components. However, linearizing along the $SO(3)$ manifold results in a singularity free parameterization which becomes particularly important during the study of highly dynamic and acrobatic motions wherein the robot may pass through points of singularity.

4.2.3 MPC Formulation

With the reduced-order model in hand, this section proceeds to formulate the MPC problem which is used as a high-level path planner for agile locomotion. The MPC formulation is

very similar to (3.18) but with a different template model and slightly different constraints. This MPC is used in order to create an optimal sequence of force inputs, f, which drives the system from some arbitrary state, x_i, to some final state, x_f, over the control horizon N_c. In particular, we consider the following optimal control problem at time step k

$$\begin{aligned}
\min_{U_{k\to k+N_c-1|k}} \quad & \mathcal{J}_k\left(\sigma_k, U_{k\to k+N_c-1|k}\right) = p\left(\sigma_{k+N_c|k}\right) + \sum_{i=0}^{N_c-1} \mathcal{L}\left(\sigma_{k+i|k}, \delta f_{k+i|k}\right) \\
\text{s.t.} \quad & \sigma_{k+i+1|k} = A_d^{\text{VBL}}\,\sigma_{k+i|k} + B_d^{\text{VBL}}\,\delta f_{k+i|k} \\
& \delta f^{\text{lb}} \le \delta f_{k+i|k} \le \delta f^{\text{ub}} \\
& f_d + \delta f_{k+i|k} \in \mathcal{FC}, \quad i = 0, 1, \cdots, N_c - 1,
\end{aligned} \tag{4.9}$$

where $U_{k\to k+N_c-1|k} := \text{col}(\delta f_{k|k}, \ldots, \delta f_{k+N_c-1|k})$ denotes the sequence of control inputs and $\sigma_{k+i|k}$ is the predicted state vector at time $k+i$ computed at time k subject to the dynamics $\sigma_{k+i+1|k} = A_d^{\text{VBL}}\,\sigma_{k+i|k} + B_d^{\text{VBL}}\,\delta f_{k+i|k}$ with the initial condition $\sigma_{k|k} := \sigma_k$. In a similar manner, we denote $\delta f_{k+i|k}$ to be the computed forces at the leg ends at time $k+i$ calculated at time k. The terminal cost is expressed as $p(\sigma_{k+N_c|k}) := ||\sigma_{k+N_c|k} - \sigma^{\text{des}}_{k+N_c|k}||^2_P$ for some symmetric positive definite matrix $P \in \mathbb{R}^{12\times 12}$. In this notation $\sigma^{\text{des}}_{k+i|k}$ represents the desired state vector at time $k+i$ which is a smooth trajectory beginning at σ_k and ending at σ_f, and $||\sigma||^2_P := \sigma^\top P \sigma$. The stage cost is also defined as $\mathcal{L}(\sigma_{k+i|k}, \delta f_{k+i|k}) := ||\sigma_{k+i|k} - \sigma^{\text{des}}_{k+i|k}||^2_Q + ||\delta f_{k+i|k}||^2_R$, where $Q \in \mathbb{R}^{12\times 12}$ and $R \in \mathbb{R}^{12\times 12}$ are symmetric positive definite matrices.

4.3 Augmented Low-Level Controller

While the planner developed in the foregoing sections has been implemented on other quadrupedal platforms, the low-level controller used is very simple. In particular, previous work has focused on tracking the forces, f_d, prescribed by the planner using the simple

mapping $\tau_r = J_c^\top f_d$. Since there is not typically a force sensor at the leg ends, the forces are tracked in an open loop manner. Conceptually, the plausibility of this scheme rests on the premise that the planner will be updated fast enough (typically at +200Hz) such that the open loop tracking is sufficient to produce stable locomotion. While this method is extremely simple and computationally efficient to implement, it is difficult to ensure adequate tracking of the forces. In this section, we examine the use of a nonlinear feedback controller, similar to that which was derived in Section 3.5, but with one key modifications in order to provide robust tracking of the desired forces and trajectory obtained from the high-level path planner. Namely, in this formulation, a Control Lyapunov Function (CLF) is added as a constraint to the convex QP.

Here we consider the same I-O linearization as defined in (3.20). The equations can be written in the form

$$\ddot{y} = \mathrm{L}_K \mathrm{L}_f y(z,t)\, \xi + \mathrm{L} y(z,t) = v, \tag{4.10}$$

where $\mathrm{L}_K \mathrm{L}_f y(z,t) := [\mathrm{L}_g \mathrm{L}_f y(z,t),\, \mathrm{L}_c \mathrm{L}_f y(z,t)]$, $\xi := \mathrm{col}(\tau_r, f)$, and $\mathrm{L} := \mathrm{L}_f^2 y(z,t) - \frac{\partial^2 y}{\partial t^2}(z,t)$. Let us define $\phi := \mathrm{col}(y,\, \dot{y})$, then (4.10) can be written in matrix form as

$$\dot{\phi} = \underbrace{\begin{bmatrix} \mathbf{0}_{n_{vc}\times n_{vc}} & I_{n_{vc}\times n_{vc}} \\ \mathbf{0}_{n_{vc}\times n_{vc}} & \mathbf{0}_{n_{vc}\times n_{vc}} \end{bmatrix}}_{F} \phi + \underbrace{\begin{bmatrix} \mathbf{0}_{n_{vc}\times n_{vc}} \\ I_{n_{vc}\times n_{vc}} \end{bmatrix}}_{G} v, \tag{4.11}$$

where n_{vc} denotes the number of virtual constraints, i.e., $n_{vc} := \dim(y)$. The nominal stabilizing controller is chosen as $v = -K_D\, \dot{y} - K_P\, y$, for some positive definite gain matrices $K_p \in \mathbb{R}^{n_{vc}\times n_{vc}}$ and $K_d \in \mathbb{R}^{n_{vc}\times n_{vc}}$ which results in stable output dynamics. The output

dynamics are then described by

$$\dot{\phi} = \underbrace{\begin{bmatrix} \mathbf{0}_{n_{vc}\times n_{vc}} & I_{n_{vc}\times n_{vc}} \\ -K_p & -K_d \end{bmatrix}}_{\hat{F}} \phi. \tag{4.12}$$

However, in the case that (4.12) cannot be satisfied exactly, a defect variable is added as follows

$$\dot{\phi} = \underbrace{\begin{bmatrix} \mathbf{0}_{n_{vc}\times n_{vc}} & I_{n_{vc}\times n_{vc}} \\ -K_p & -K_d \end{bmatrix}}_{\hat{F}} \phi + \underbrace{\begin{bmatrix} \mathbf{0}_{n_{vc}\times n_{vc}} \\ I_{n_{vc}\times n_{vc}} \end{bmatrix}}_{G} \hat{v}. \tag{4.13}$$

This is equivalent to the outputs for the low-level controller defined in (3.5). The previous derivation attempts to minimize $\hat{v}$, but does so in a blind manner. Here, we aim to develop a controller which may still deviate from the nominal control input, but will do so in a manner that ensures the output dynamics remain Lyapunov stable [85, 141]. Using the ideal controller (i.e., the case where $\hat{v} = \mathbf{0}$) and the Algebraic Lyapunov Equation

$$\hat{F}^\top P + P\hat{F} = -S,$$

we may solve for P, where S is assumed to be the identity matrix. We may then define the candidate Lyapunov function as $V := \phi^\top P \phi$ whose derivative is given by

$$\dot{V} = \dot{\phi}^\top P\phi + \phi^\top P\dot{\phi} \tag{4.14}$$

$$= (\hat{F}\phi)^\top P\phi + (G\hat{v})^\top P\phi + \phi^\top P(\hat{F}\phi) + \phi^\top P(G\hat{v}) \tag{4.15}$$

$$= \underbrace{\phi^\top(\hat{F}^\top P + P\hat{F})\phi}_{L_F V} + \underbrace{2\,\phi^\top PG}_{L_G V}\hat{v} \tag{4.16}$$

where $L_F V := -\phi^\top S\,\phi$ and $L_G V := 2\,\phi^\top PG$. In order to obtain a rapidly exponentially stable CLF, we may write

$$\dot{V}_\epsilon \leq -\frac{c}{\epsilon} V_\epsilon, \tag{4.17}$$

where $c = \frac{\lambda_{\min}(S)}{\lambda_{\max}(P)}$, $0 < \epsilon < 1$, and

$$V_\epsilon := \phi^\top \begin{bmatrix} \frac{1}{\epsilon} & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} \frac{1}{\epsilon} & 0 \\ 0 & I \end{bmatrix} \phi.$$

In this notation ϵ is used as a tuning variable for the rate of convergence. For compact notation, let us define

$$\psi_0 := L_F V_\epsilon + \frac{c}{\epsilon} V_\epsilon$$
$$\psi_1 := L_G V_\epsilon.$$

Now we may modify the QP as follows:

$$\min_{(\tau_r, \hat{v}, f, \delta)} \quad \frac{\Gamma_\tau}{2}\tau_r^\top \tau_r + \frac{\Gamma_{\hat{v}}}{2}\hat{v}^\top \hat{v} + \frac{\Gamma_f}{2}(f - f_d)^\top (f - f_d) + \frac{\Gamma_\delta}{2}\delta^2$$

$$\begin{aligned}
\text{s.t.} \quad & \mathrm{L}_K \mathrm{L}_f\, \xi + \mathrm{L} = -K_p y - K_d \dot{y} + \hat{v} && \text{(I-O Linearization)} \\
& J_c D_r^{-1}(B_r \tau_r + J_c^\top f - H_r) + \dot{J}_c = 0 && \text{(Zero Acceleration at Stance Leg Ends)} \\
& \psi_0 + \psi_1 \hat{v} \leq \delta && \text{(Relaxed CLF Condition)} \\
& f \in \mathcal{FC} && \text{(Friction Cone Condition)} \\
& \tau_r^{\mathrm{lb}} \leq \tau_r \leq \tau_r^{\mathrm{ub}} && \text{(Feasibility of Torques)} \\
& \hat{v}^{\mathrm{lb}} \leq \hat{v} \leq \hat{v}^{\mathrm{ub}} && \text{(Feasibility of the Auxiliary Input),}
\end{aligned} \tag{4.18}$$

where each $\Gamma_{(\cdot)}$ is a positive scalar used for tuning, $\mathcal{FC}$ is the friction cone, and $(\cdot)^{\text{ub}}$ and $(\cdot)^{\text{lb}}$ represent upper and lower bounds, respectively. Note that in this formulation, there is one additional slack variable, δ, added on the CLF condition. This is to ensure feasibility of the QP in the case where the CLF constraint cannot be satisfied. Similar to the previous formulation, a much higher weight is placed on the slack variable to drive it as close as possible to zero. Adding the CLF to the QP formulation only adds one additional constraint (since the CLF is a scalar value function) and one decision variable (the slack variable). However, the addition of the CLF allows one to relax the bound constraints on $\hat{v}$ which conditions the QP better, and in turn reduces the number of iterations that it takes to reach an optimal solution.

Unlike the previous formulation, this QP also considers desired forces produced by the high-level MPC. The addition of the desired forces allows for more stable motion and better tracking during challenging motions, particularly when traversing rough terrain and moving at a high rate of speed. Since the COM trajectory for the quadruped is still operating on a heuristic trajectory, the forces provided by the MPC condition the QP in a manner such that better overall tracking is achieved. Specifically, the addition of well conditioned desired forces creates a gait that is significantly more stable in the medial/lateral direction due to the relaxation on the I-O linearization.

4.4 Numerical Simulations

The objective of this section is to display the efficacy of both the high-level planner and low-level nonlinear controller developed in this chapter. In particular, we present a series of full-order simulations using the A1 quadruped to demonstrate the ability for this hierarchical control scheme to produce high speed and robust locomotion in the presence of uncertainty

in ground height and external disturbances. In all cases provided here, a trot gait is considered. All of the full-order simulations here are produced using RaiSim [9], where we utilize qpSWIFT [134] to solve the high- and low-level QPs. The MPC problem (4.9) is solved at 200Hz, whereas the low-level nonlinear control QP (4.18) is solved at 1kHz. We can show that the QP for the higher-level MPC has 240 decision variables for all domains, regardless of the number of ground contacts. In addition, the typical computation time of the MPC problem on a desktop computer with an Intel® Xeon® W-2125 CPU at 4.00 GHz (4 cores) and 16 GB of RAM is 0.576ms. The lower-level QP has 40 decision variables for both the double- and quadruple-contact domains of the trot gait with an average typical computation time of approximately 0.131ms.

4.4.1 Agile Locomotion

In this section we consider the simulation results for nominal locomotion. Using the control approach outlined above, the quadruped is able to perform inplace, forward, lateral, and diagonal trots over a wide range of speeds in a stable manner. Here, there are no external disturbances and the ground is assumed to be flat. Under these conditions, the quadruped may reach forward velocities as high as ~ 3.1 (m/s) which is comparable to the velocities that may be achieved using the manufacturers built in gait. After this point the robot is largely limited by joint and torque restrictions which prevent the robot from feasibly traveling faster under this choice of control. Furthermore, the lateral stability of the quadruped begins to become questionable and exhibits oscillatory behavior when moving this fast. It should be noted, however, that these speeds are seldom reached for legged robots even in simulation.

Here we aim to show four sets of simulations. The outputs (i.e., virtual constraints) and torques are provided in Fig. 4.3 for a forward, backward, lateral, and diagonal trot. In the

case of the forward and backwards trots (Fig. 4.3 (a) and (b), respectively), the commanded velocity is 3 (m/s). The lateral trot is shown in Fig. 4.3 (c), and the diagonal trot is shown in Fig. 4.3 (d), both with a commanded velocity of 1 (m/s). In a similar manner, plots showing the time-response of the CLF are shown in Fig. 4.4 (a)-(d). It is evident that the CLF constraint is respected throughout locomotion, meaning that the defect variable placed on the virtual constraints are determined in manner which is consistent with rapidly exponentially stable outputs. In particular, V remains positive, and $\dot{V}$ remains negative.

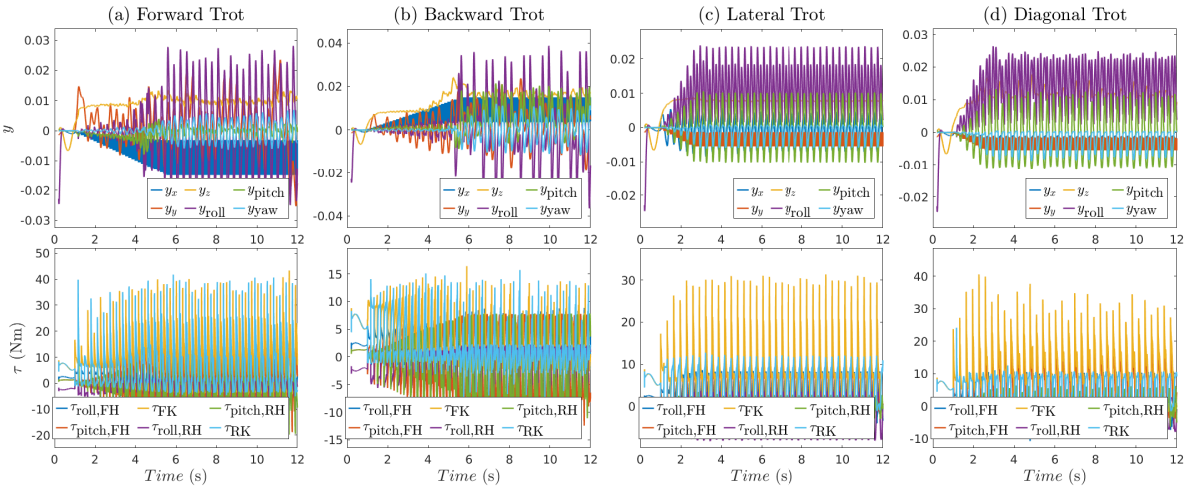

Figure 4.3: Torques and virtual constraints for nominal trots in various directions. (a) and (b) depict forward and backward trots, respectively, each with a command velocity of 3 (m/s). (c) shows a lateral trot with a command velocity of 1 (m/s), and finally in (d) we provide a diagonal trot with a command velocity of 1 (m/s) in both the forward and lateral directions. Here, we only plot the components of the virtual constraints that correspond to the absolute position of the COM (i.e., y_x, y_y, and y_z) and absolute orientation of the robot's floating base (i.e., y_{roll}, y_{pitch}, y_{yaw}). For the torque plots, the subscripts "FH", "FK", "RH", and "RK" stand for the front hip, front knee, rear hip, and rear knee of the left-hand-side of the robot, respectively. In addition, "roll" and "pitch" for the torque plots represent the roll and pitch motions of the joints.

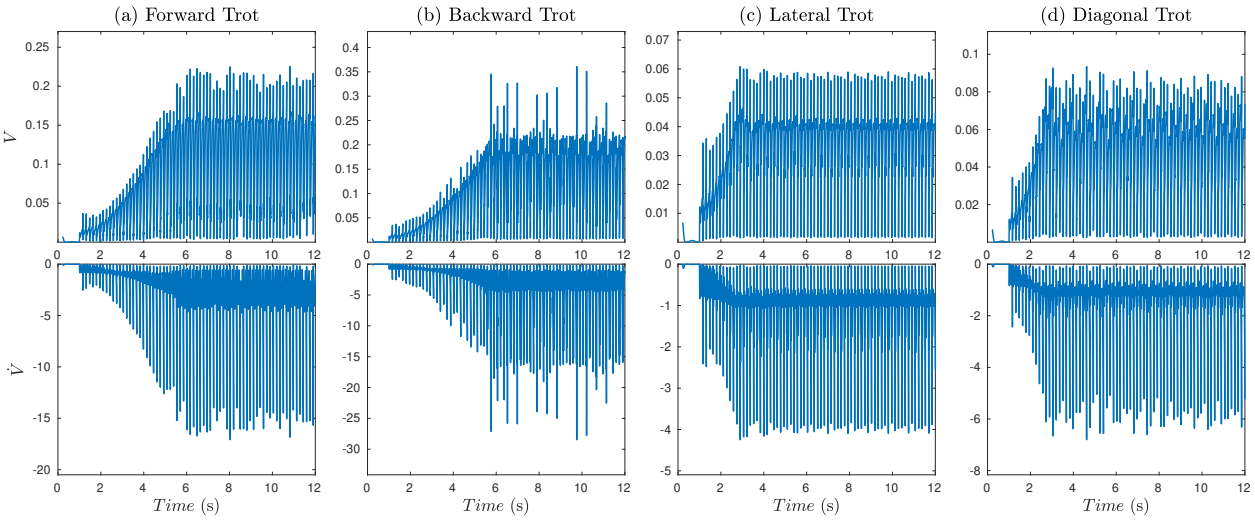

Figure 4.4: The Lyapunov function for the simulations shown in Fig. 4.3.

4.4.2 Robust Locomotion

Blind Locomotion Along Rough Terrain

In order to determine the robustness of the system to unknown terrain, a series of simulations were conducted in which the quadruped is required to navigate varying terrain in a blind manner with *no* prior knowledge of the terrain (i.e., the controller operates under the assumption that the ground is flat). In these simulations, blocks of randomly varying height are placed in the path of locomotion, where the blocks on the left and right side of the robot may differ (i.e., introducing medial/lateral asymmetry in the terrain), and the blocks vary in height along the direction of locomotion (see Fig. 4.5 for a snapshot of the quadruped navigating this terrain). Here we consider four different scenarios, all of which use a trot gait. In the first two scenarios, the quadruped is commanded to perform a forward trot with velocities of 1 (m/s) and 2 (m/s), where the outputs and torques for these simulations can be found in Fig. 4.6 (a) and (b), respectively. Furthermore, the maximum allowable ground height that for these simulations are 12 (cm) and 8 (cm) for the fast and slow trots, respec-

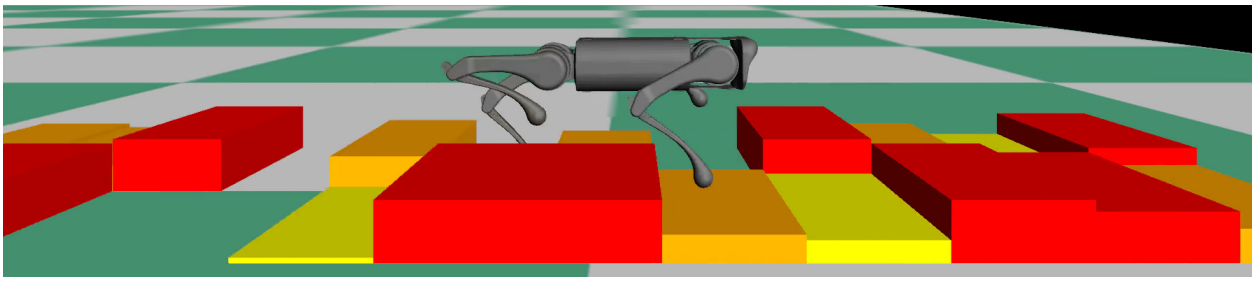

Figure 4.5: Example of the what the rough terrain could look like. In this figure, the blocks range from a height of 0 (cm) to 12 (cm) (40% of the A1 robot's height) in a random manner.

tively. In the latter simulations, we consider a backward trot and a lateral trot, each with a commanded velocity of 1 (m/s), where the maximum ground height variations are 12 (cm) and 8 (cm), respectively. The outputs and torques for these simulations can be found in Fig. 4.6 (c) and (d). In an analogous manner, the Lyapunov function for all of these simulations may be found in Fig. 4.7. Finally, a series of snapshots corresponding to Fig. 4.6 (a) can be found in Fig. 4.8. From these simulations it is evident that under this choice of control the quadruped is able to navigate ground height disturbances of a significant magnitude in a blind and stable manner. *In fact, the maximum allowable height used in the simulations provided here (*8 *(cm) max and* 12 *(cm) max) are more than* 28% *and* 40% *of the commanded standing height, respectively.* This shows that the control scheme detailed in this work is highly robust to ground height variations even during blind locomotion.

Push Recovery

The primary purpose of this section is to examine the robustness of this controller in the presence of external disturbances. Unlike in the quasi-static case, here the robustness comes strictly from the hierarchical locomotion controller since the tail is not used in the current formulation. Four simulations are presented, wherein the first two include a pulse-like disturbance which lasts for 200ms, and the second two involve persistent disturbances. In all

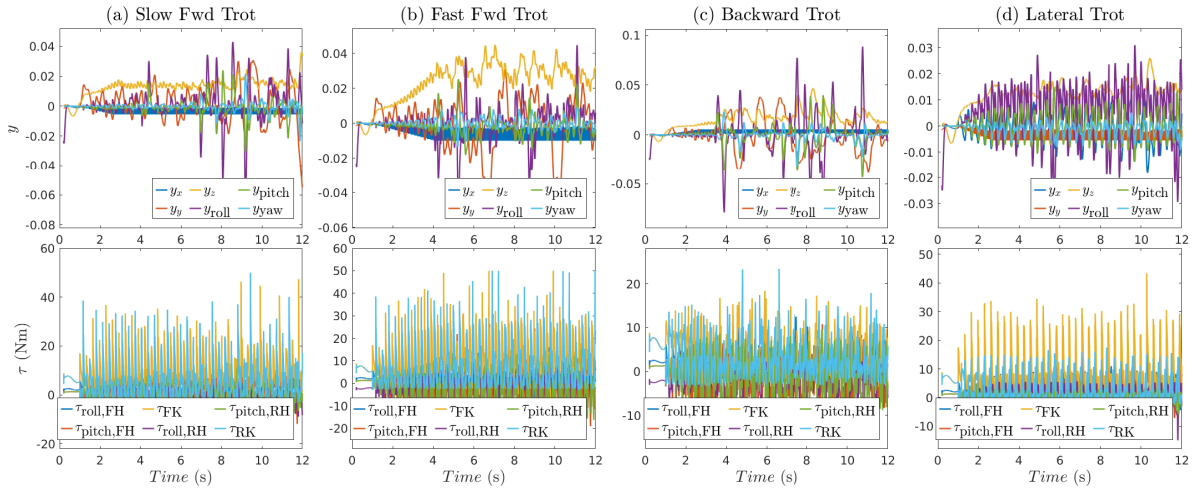

Figure 4.6: (a) and (b) show the virtual constraints and the torques for forward trots over rough terrain. In (a), the commanded velocity is 1 (m/s), and in (b) the commanded velocity is 2 (m/s). (c) and (d) show the virtual constraints and torques for a backward trot and lateral trot, each with a command velocity of 1 (m/s). In (a) and (c), the blocks take heights in the range $[0, 12]$ (cm), while in (b) and (d) the blocks range in height according to $[0, 8]$ (cm).

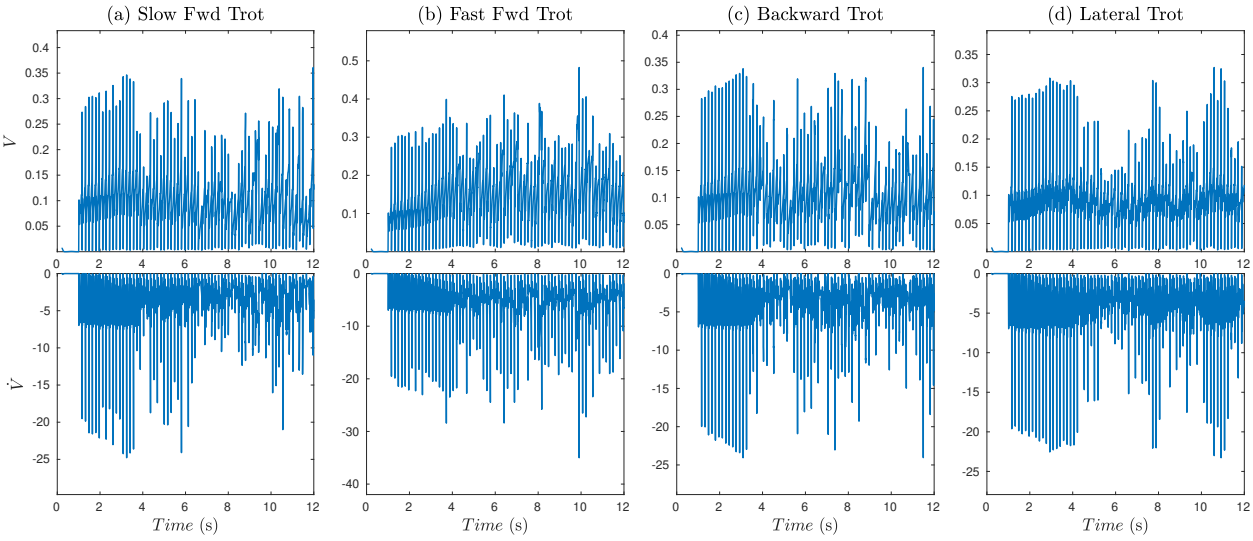

Figure 4.7: The Lyapunov function for the simulations shown in Fig. 4.6.

cases, the quadruped is performing a forward trot with a commanded velocity of 1 (m/s). Fig. 4.9 (a) shows the system after having undergone a forward pulse with a magnitude of 120 (N), while (b) shows locomotion perturbed by a lateral pulse of 40 (N). Fig. 4.9 (c) and

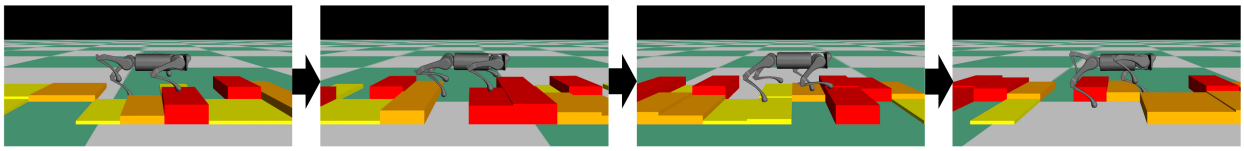

Figure 4.8: Snapshots of the simulations of the A1 robot found in Fig. 4.6 (a).

(d) show locomotion with a persistent disturbance of the form $\beta \sin(4t)$ (N) and operate in the anterior/posterior and medial/lateral directions, respectively. The term β is set to be $\beta = 50$ and $\beta = 20$ for (c) and (d), respectively. In addition, the plots of the CLF for each simulation are shown in Fig. 4.10. In all cases, the robot is able to adequately perform under these conditions without falling which further shows the robustness of this control scheme.

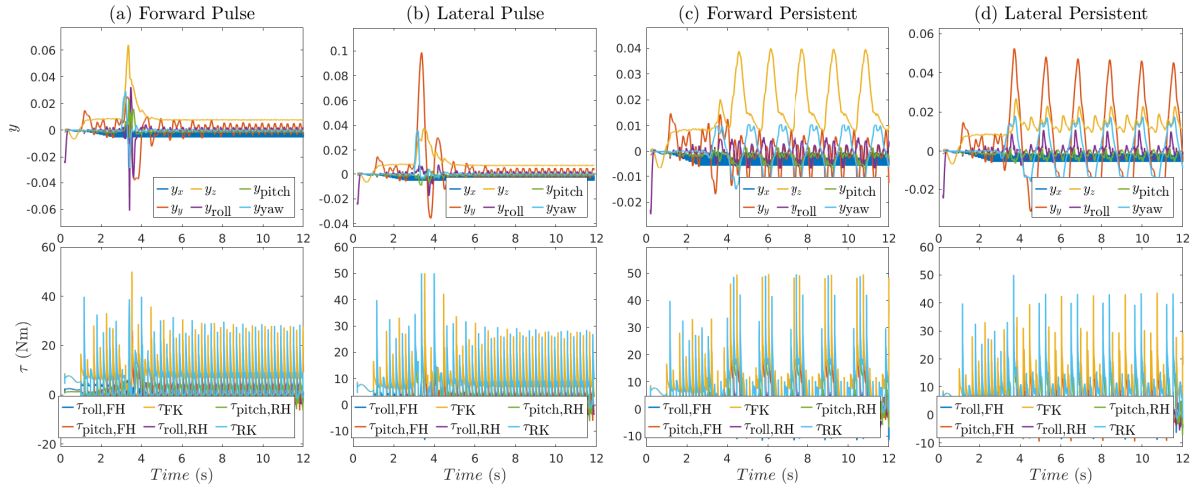

Figure 4.9: Torques and virtual constraints for a forward trot at 1 (m/s) subject to different external disturbances applied at the COM. (a) and (b) depict simulations of a trot which has a pulse disturbance lasting for 200ms in the forward and lateral directions with magnitudes of 120 (N) and 40 (N), respectively. (c) and (d) show the same trot subject to a sinusoidal disturbance in the forward and lateral directions according to $50 \sin(4t)$ and $20 \sin(4t)$, respectively.

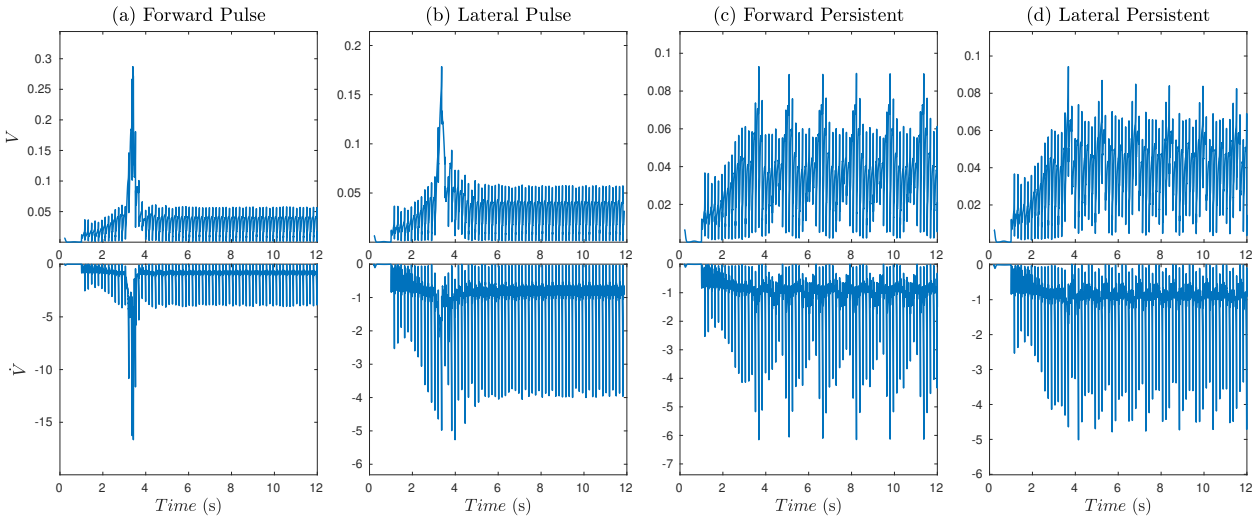

Figure 4.10: The Lyapunov function for the simulations shown in Fig. 4.9.

4.5 Preliminary Experimentation

The primary goal of this section is to provide experimental validation of the low-level controller proposed in Chapter 4. There have been very few fully nonlinear low-level controllers used for the implementation of quadrupedal locomotion (i.e., without approximations), and even fewer which consider a CLF for stability. Furthermore, this is the first time that hardware experiments have been conducted to implement nonlinear feedback linearization without approximations. For this reason, the MPC has been omitted for preliminary experiments and we focus solely on experimentally validating the optimal nonlinear feedback linearization.

For the experimental results, we consider the use of the A1 quadruped. However, this particular robot does not have sufficient computational resources onboard, so the torque is computed using an external laptop and the commands are then streamed to the robot via TCP/IP. The control loop (and communication) between the onboard micro processing unit (MCU) and the laptop occurs at 1kHz, with a round trip delay of approximately 8ms. This delay was briefly evaluated in simulation and did not appear to cause excessive performance

degradation, and does not noticeably affect the experimental results. On the external laptop, the average computation time over one contact domain for the low-level QP is 0.2228ms, where the computer is equipped with an 11^{th} gen Intel® Core(TM) i7-1185G7 running at 3.00 GHz and 16 GB of RAM.

All joint positions are available from the SDK provided with the robot, though some of the required measurements are either unavailable (COM position and velocity) or are poor enough (joint velocities) that filtering and estimation are required before implementation can be achieved. In particular, the estimates on the position and velocity of the COM are calculated using the forward kinematics of the robot paired with assumptions on the contacting leg ends. The velocity of the COM is estimated in a similar manner using the Jacobians of the contacting legs. While this has worked well, erroneous results may occur when the assumptions regarding the position of the stance legs are broken. Therefore, this method will be altered in the future to allow for more robust estimation of the position and velocity of the COM in the global coordinate frame wherein the stance feet may not be at a height of zero and small amounts of foot slippage may occur. The joint level velocities provided are extremely noisy and are not amenable to control. For that reason, the joint velocities are calculated using the joint position measurements paired with a derivative Savitzky-Golay least-squares filter [142] that significantly reduces noise and, as will be shown, produces very good results during experimentation.

4.5.1 Pose Control

In this section, a variety of experimental and simulated data are shown together for different pose control trajectories. In this case, the term pose control is used to indicate that all four legs are in contact with the ground and a trajectory is commanded for the COM position

and orientation. Six different sets of plots are provided, where the different pose scenarios consist of motion in the x (Fig. 4.11-4.13), y (Fig. 4.14-4.16), and z (Fig. 4.17-4.19) directions, as well as rotation in the roll (Fig. 4.20-4.22), pitch (Fig. 4.23-4.25), and yaw (Fig. 4.26-4.28) directions. For each pose, the outputs, torques, and Lyapunov functions for both the experimental and simulated data are provided. Each different pose consists of a sinusoidal trajectory of the form $\beta \sin(0.8\pi\, t)$. In particular, the positional poses consist of $\beta := \{0.04, 0.04, 0.05\}$ (m) for the x, y, and z directions, respectively. The rotational poses consist of $\beta := \{0.3491, 0.17453, 0.17453\}$ (rad) in the roll, pitch, and yaw directions, respectively. For each pose scenario, only one floating base state is commanded to follow a trajectory while all of the other states are commanded to remain constant, therefore isolating one particular motion at a time.

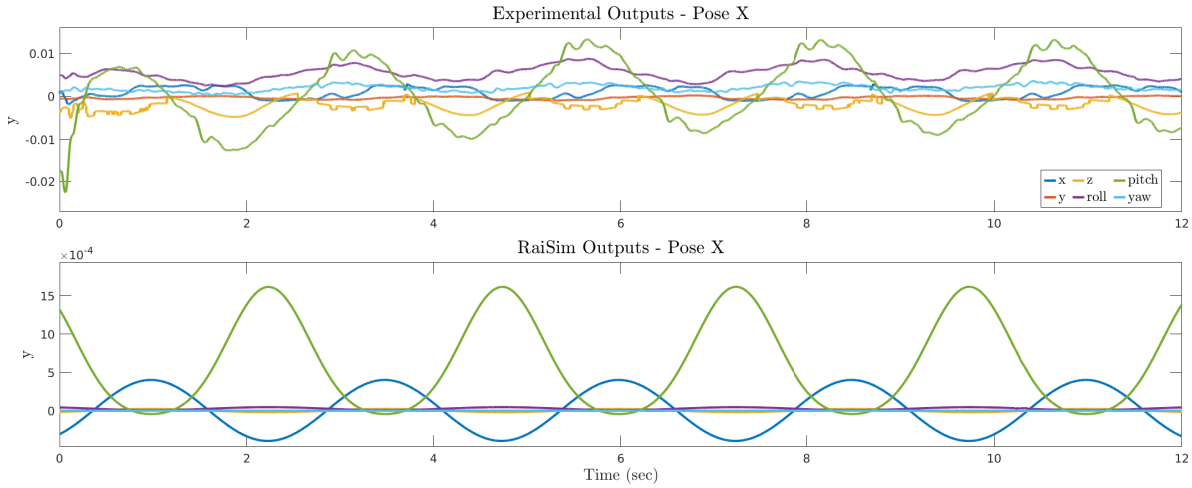

Figure 4.11: The simulated and experimental outputs (i.e., virtual constraints) for pose control in the x direction. The COM is commanded to move according to $0.04 \sin(0.8\pi\, t)$ (m)

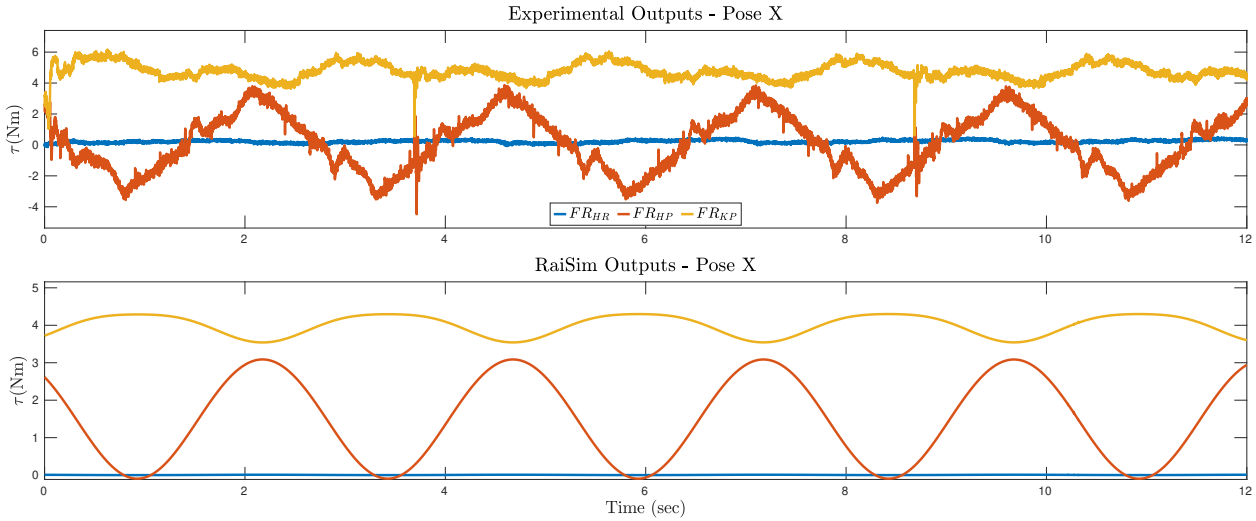

Figure 4.12: The simulated and experimental torques for the front right leg corresponding to Fig. 4.11.

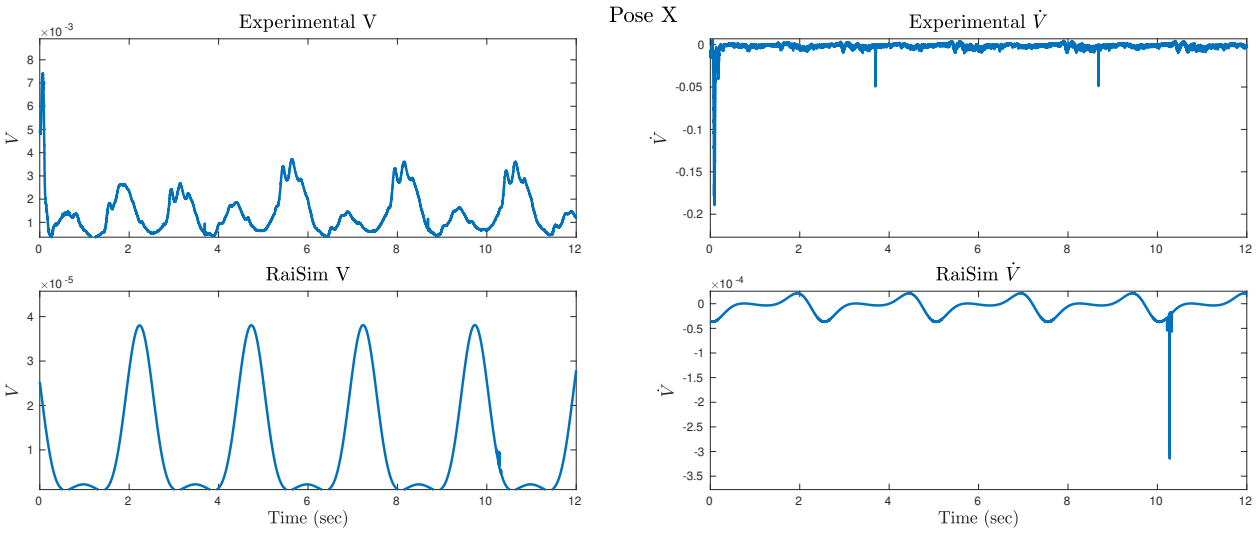

Figure 4.13: The simulated and experimental Lyapunov function corresponding to Fig. 4.11.

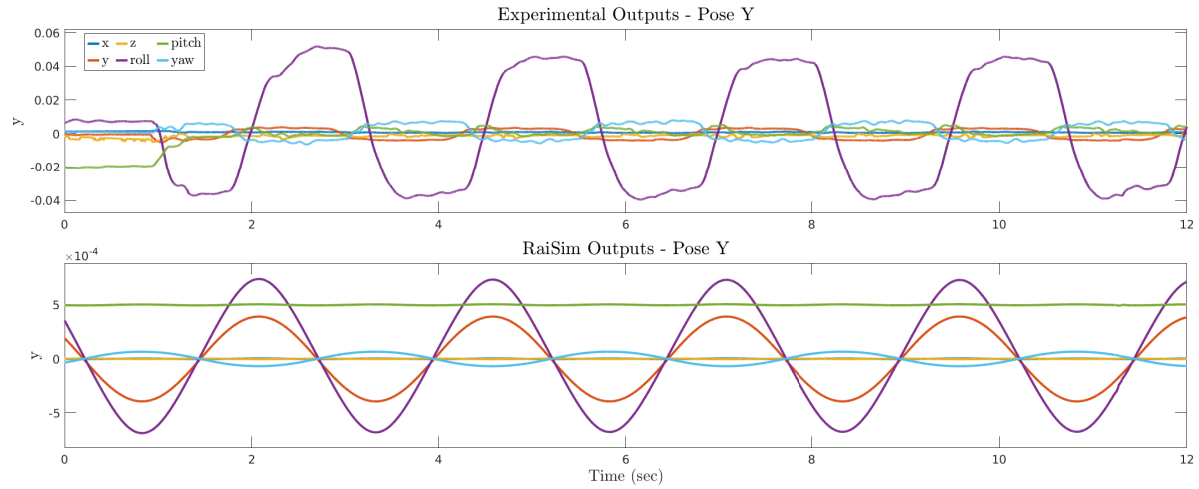

Figure 4.14: The simulated and experimental outputs (i.e., virtual constraints) for pose control in the y direction. The COM is commanded to move according to $0.04\sin(0.8\pi\, t)$ (m).

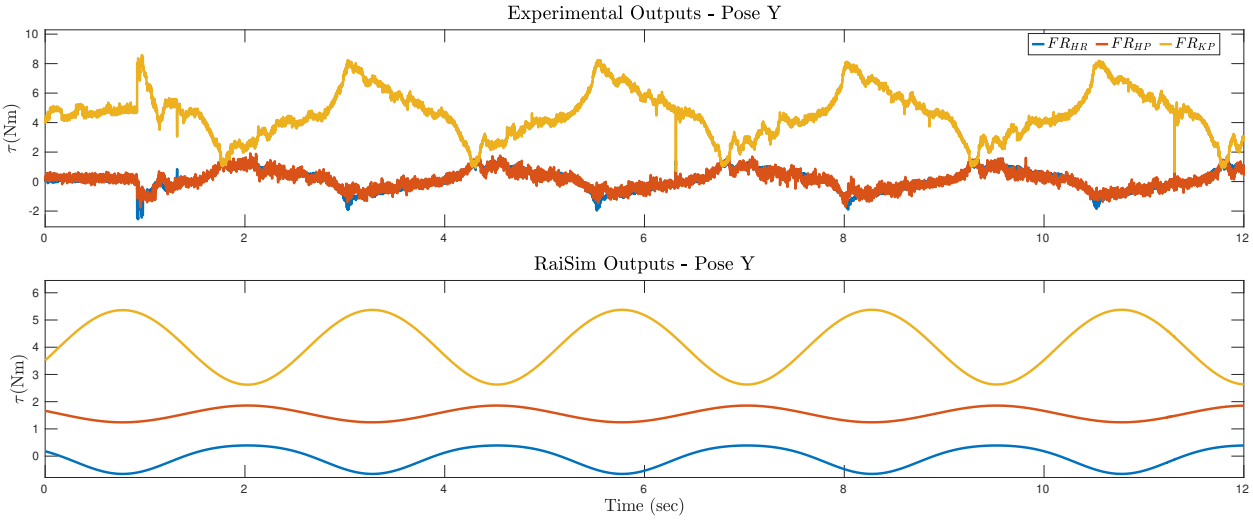

Figure 4.15: The simulated and experimental torques for the front right leg corresponding to Fig. 4.14.

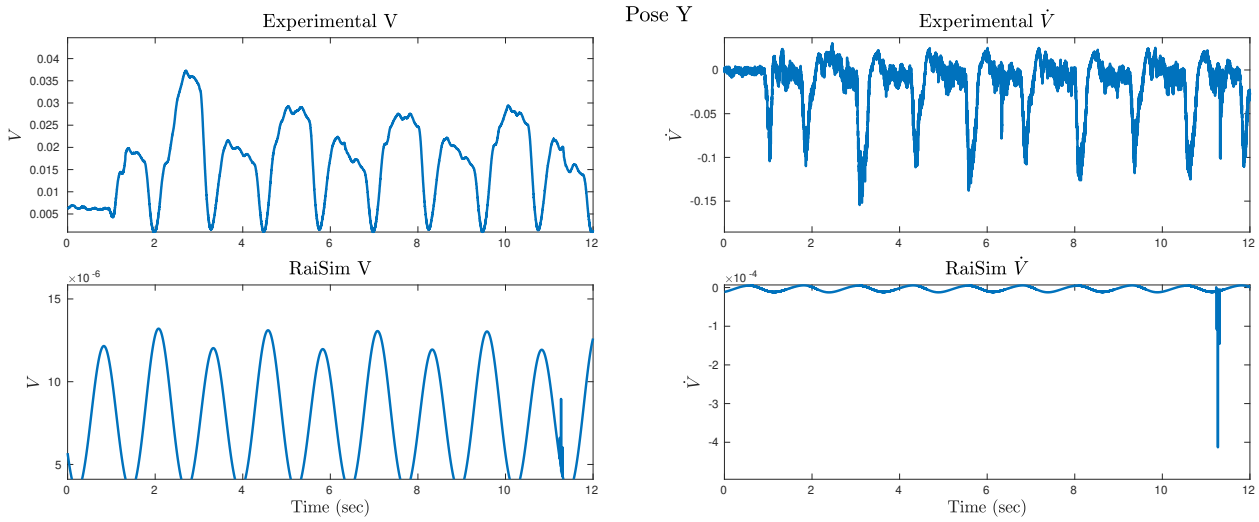

Figure 4.16: The simulated and experimental Lyapunov function corresponding to Fig. 4.14.

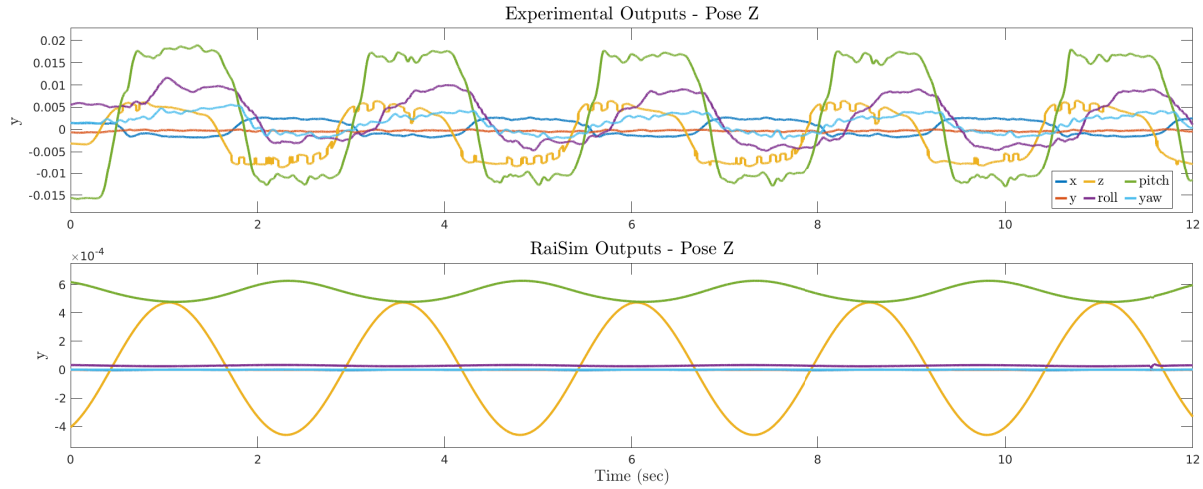

Figure 4.17: The simulated and experimental outputs (i.e., virtual constraints) for pose control in the z direction. The COM is commanded to move according to $0.05\sin(0.8\pi\, t)$ (m).

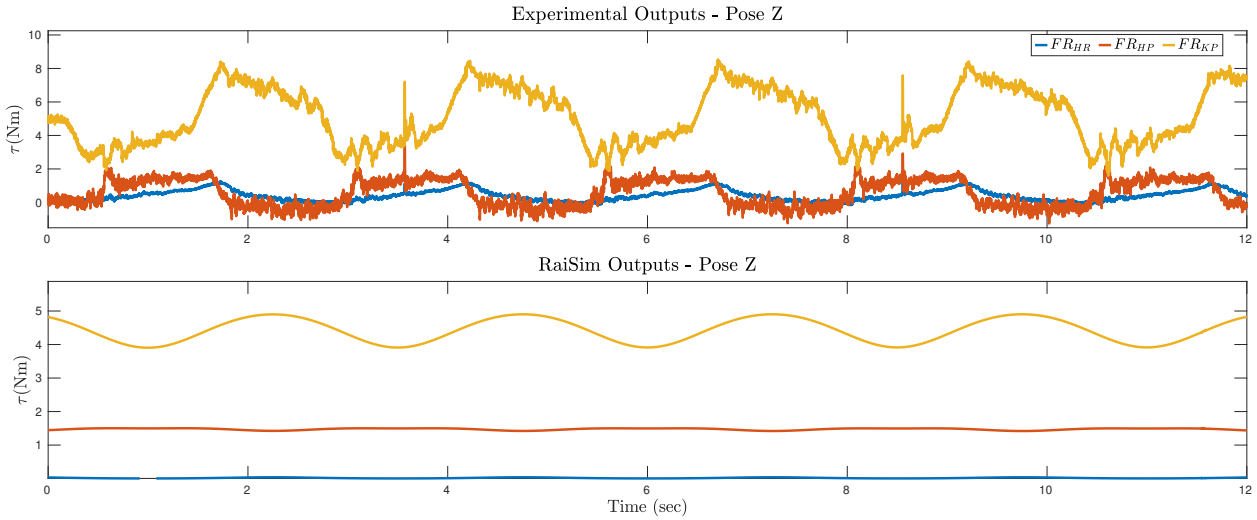

Figure 4.18: The simulated and experimental torques for the front right leg corresponding to Fig. 4.17.

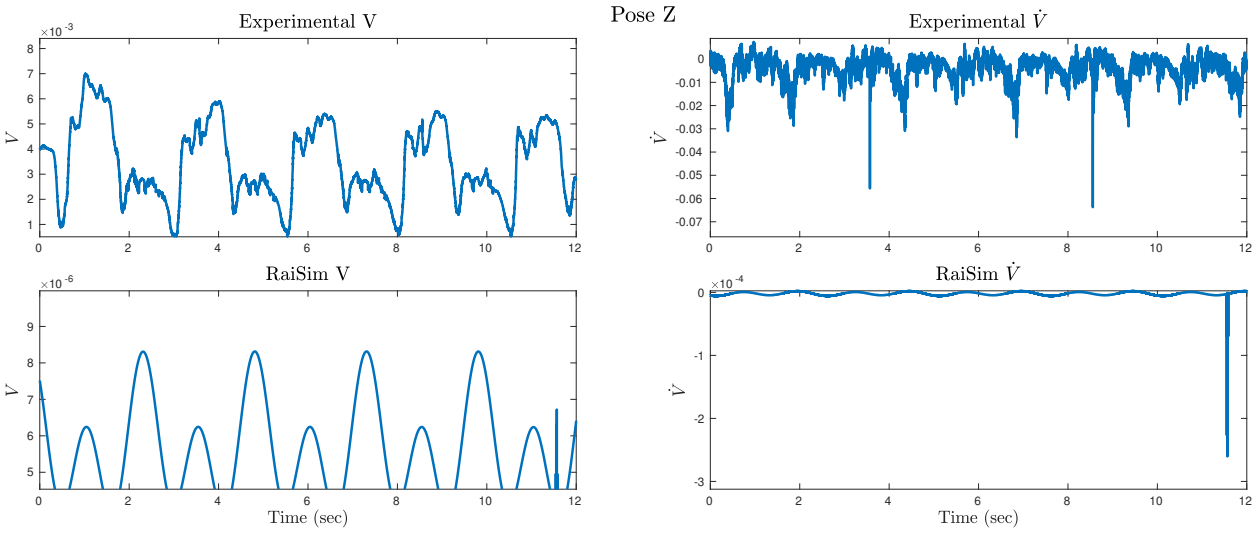

Figure 4.19: The simulated and experimental Lyapunov function corresponding to Fig. 4.17.

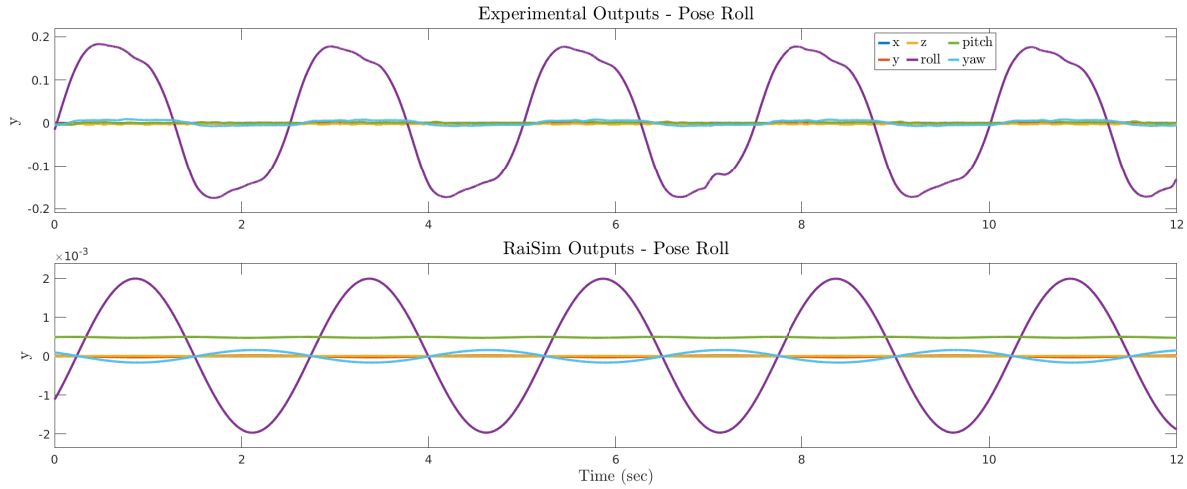

Figure 4.20: The simulated and experimental outputs (i.e., virtual constraints) for pose control in the roll direction. The COM is commanded to rotate according to $0.3491 \sin(0.8\pi\, t)$ (rad).

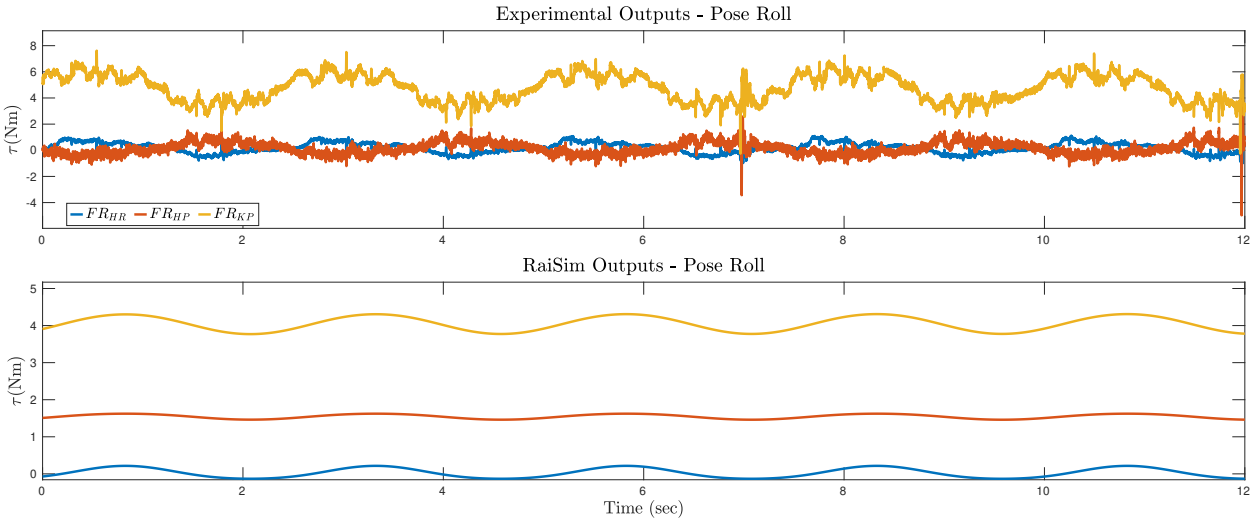

Figure 4.21: The simulated and experimental torques for the front right leg corresponding to Fig. 4.20.

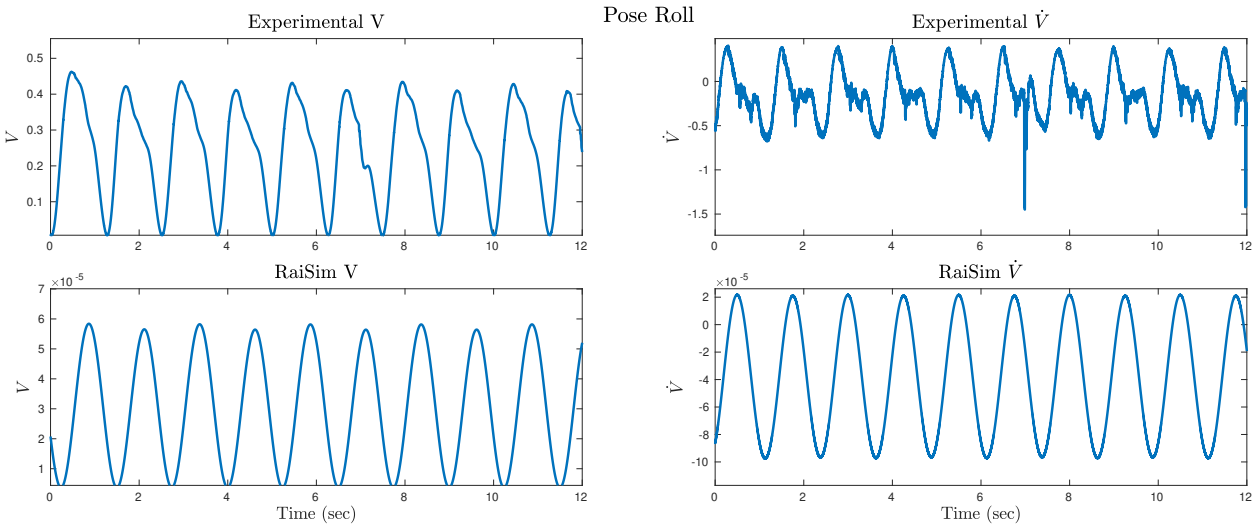

Figure 4.22: The simulated and experimental Lyapunov function corresponding to Fig. 4.20.

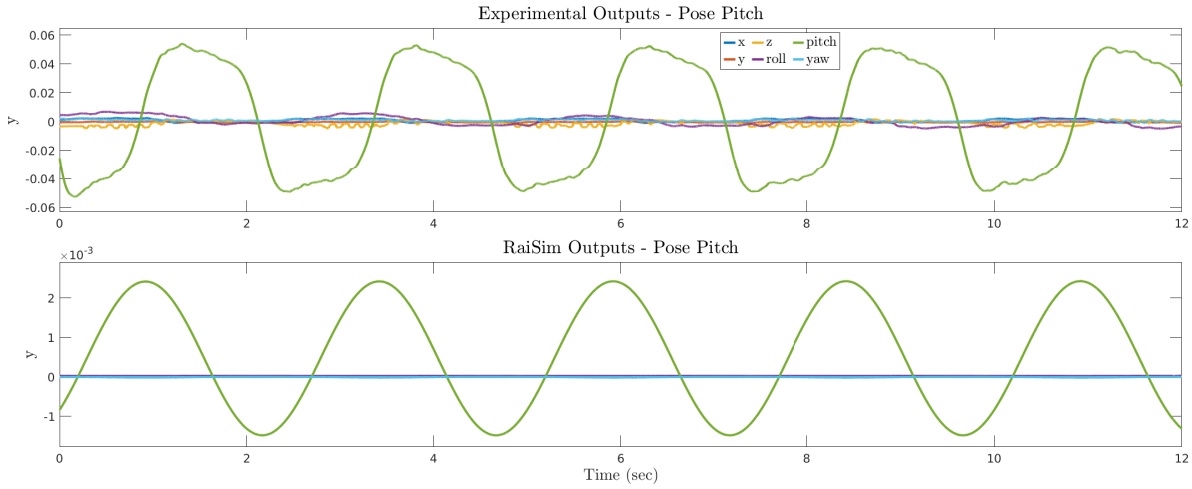

Figure 4.23: The simulated and experimental outputs (i.e., virtual constraints) for pose control in the pitch direction. The COM is commanded to rotate according to $0.17453 \sin(0.8\pi\, t)$ (rad).

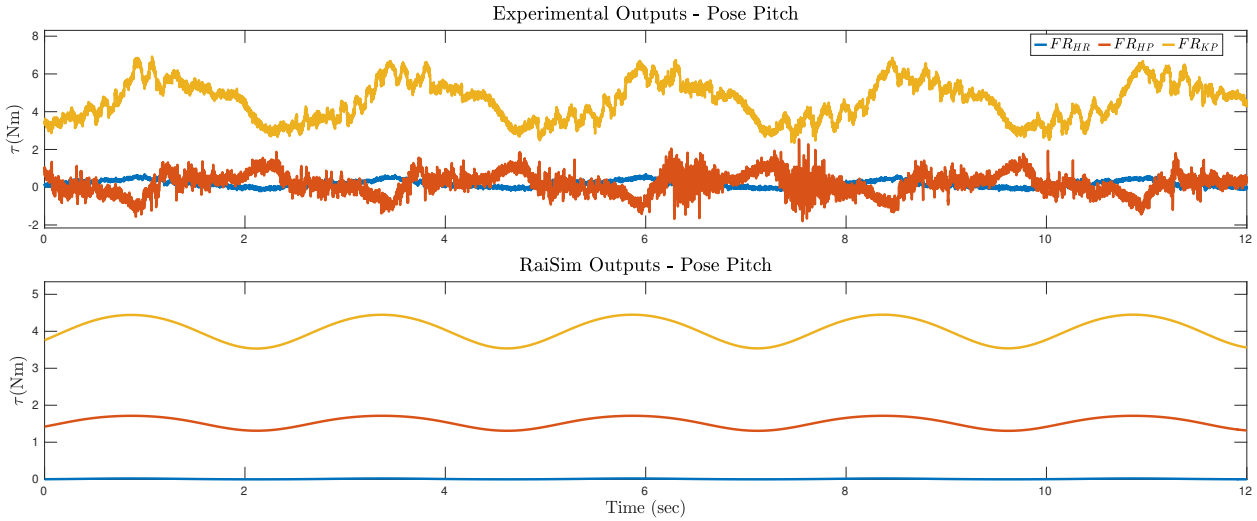

Figure 4.24: The simulated and experimental torques for the front right leg corresponding to Fig. 4.23.

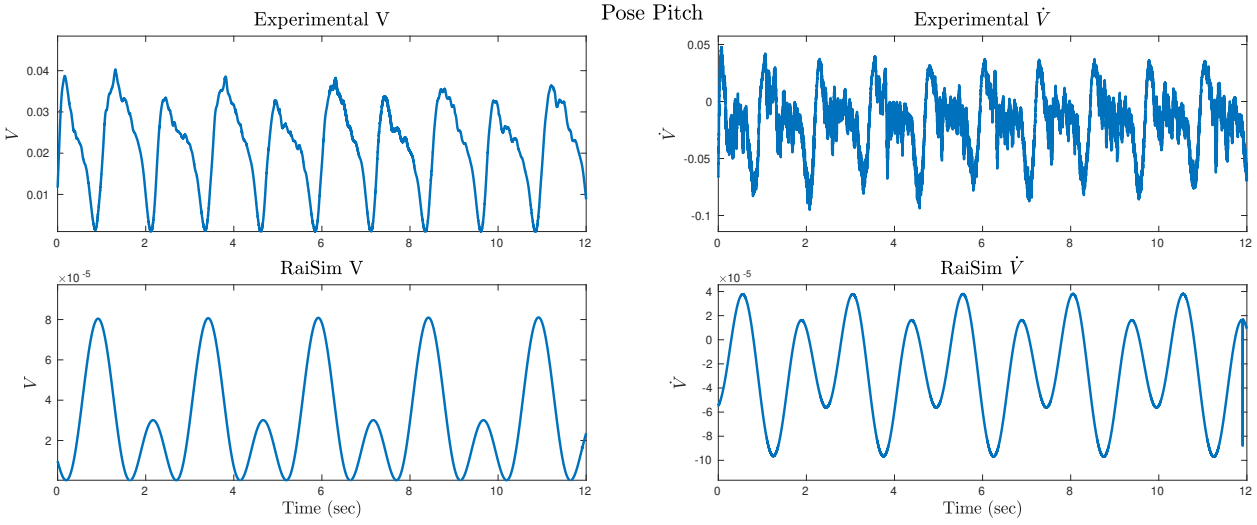

Figure 4.25: The simulated and experimental Lyapunov function corresponding to Fig. 4.23.

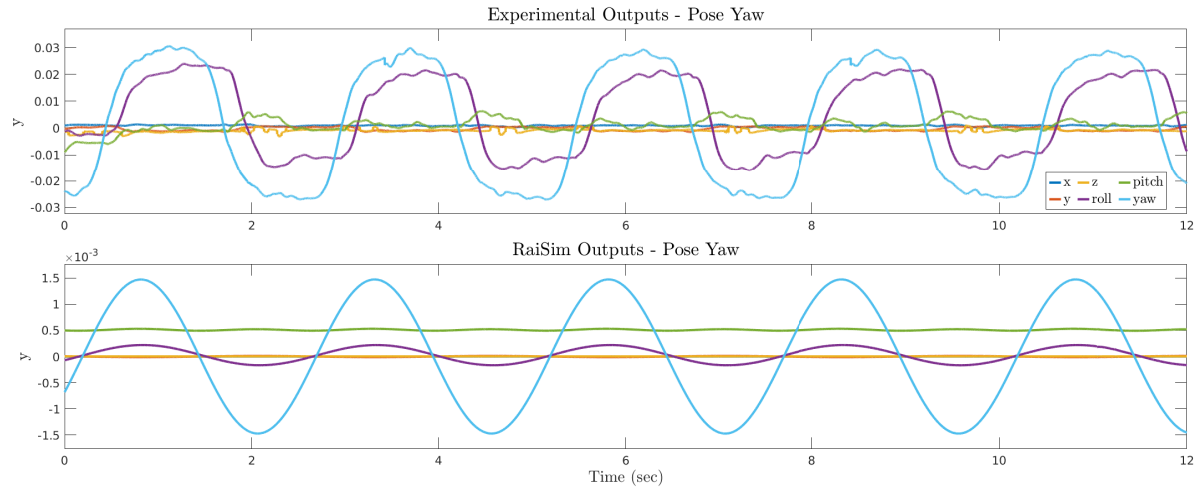

Figure 4.26: The simulated and experimental outputs (i.e., virtual constraints) for pose control in the pitch direction. The COM is commanded to rotate according to $0.17453 \sin(0.8\pi\, t)$ (rad).

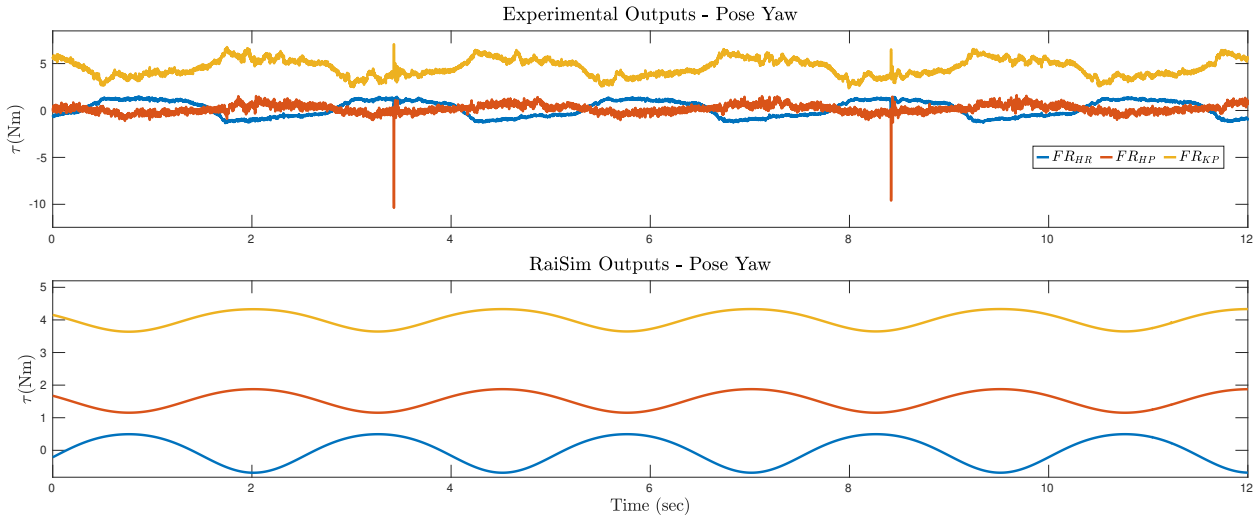

Figure 4.27: The simulated and experimental torques for the front right leg corresponding to Fig. 4.26.

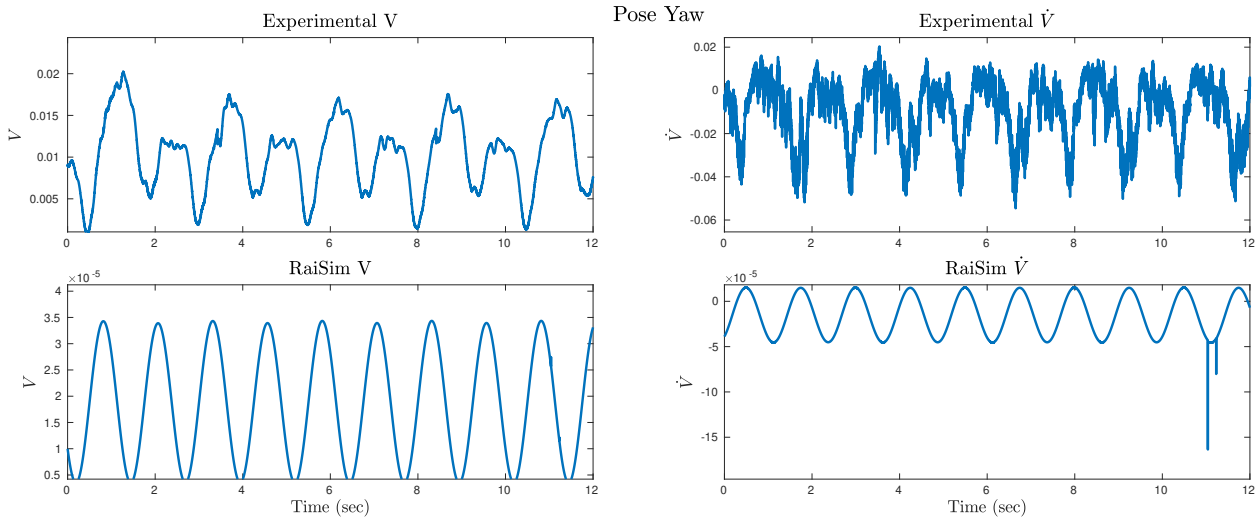

Figure 4.28: The simulated and experimental Lyapunov function corresponding to Fig. 4.26.

4.5.2 Trotting

The last set of experiments correspond to the quadruped trotting over flat, even terrain. We begin with an in-place trot, wherein the quadruped is commanded to have zero velocity. The outputs, torques, and Lyapunov function for the experimental and simulated in-place trot can be found in Figs. 4.29-4.31. In a similar manner, Figs. 4.32-4.34 correspond to the results where the quadruped is commanded to walk forward with a velocity of 0.1 (m/s). Snapshots of the experimental forward trot can be found in Fig. 4.35. The results for the backward trot with a commanded velocity of -0.1 (m/s) can be found in Figs. 4.36-4.38. Finally, in Figs. 4.39-4.41 a lateral trot is considered with a command velocity of -0.2 (m/s). Snapshots of the experimental lateral trot can be found in Fig. 4.42. In all cases, stable locomotion is achieved using only the low-level controller, though it is apparent that the simulated case produces much smoother results, likely due to model uncertainty and sensor noise.

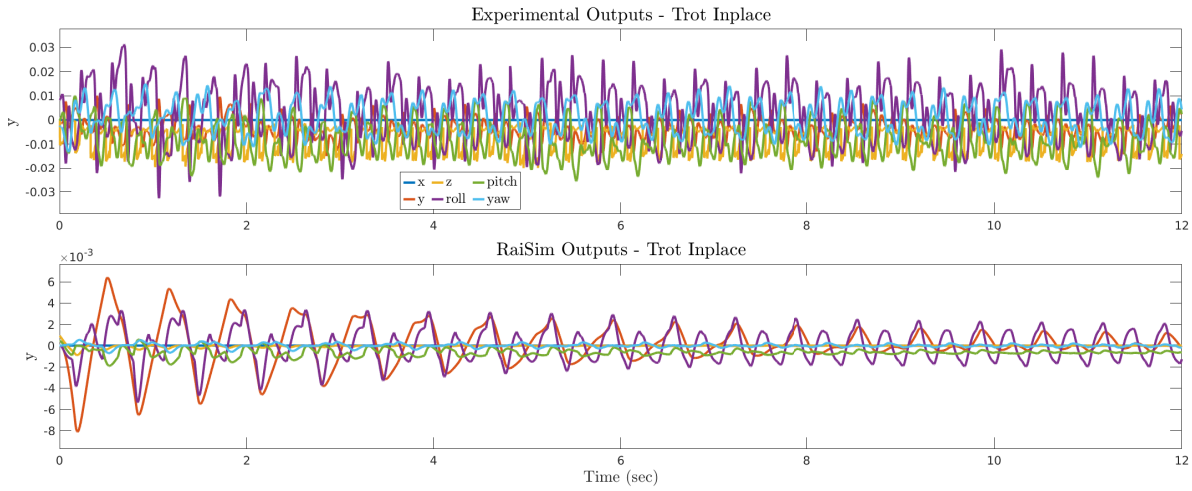

Figure 4.29: The simulated and experimental outputs (i.e., virtual constraints) for an in-place trot with a command velocity of 0.0 (m/s).

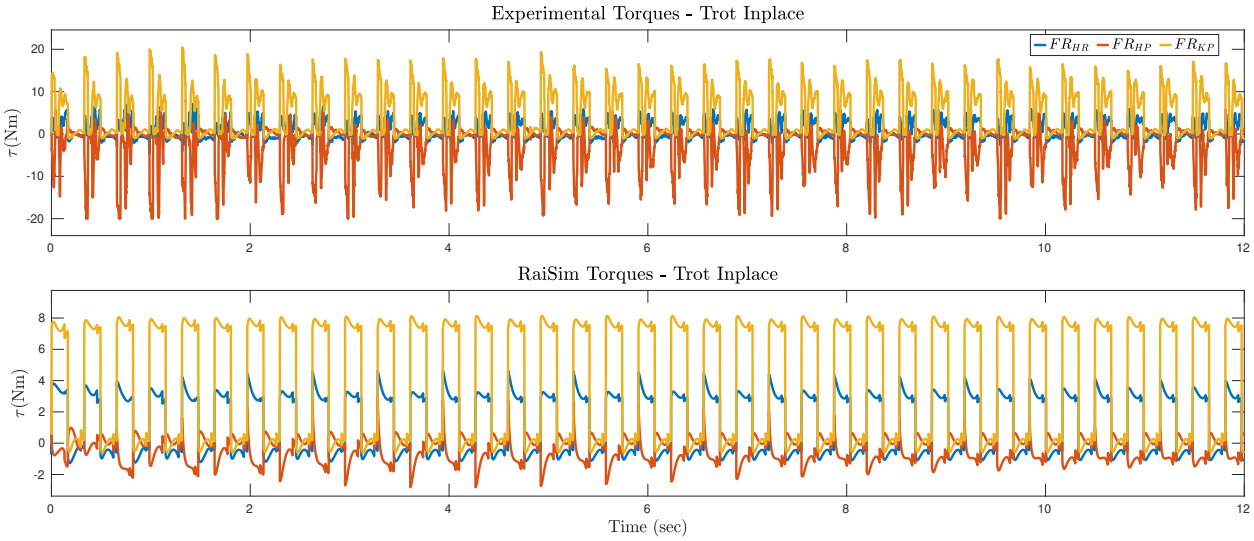

Figure 4.30: The simulated and experimental torques for the front right leg corresponding to Fig. 4.29.

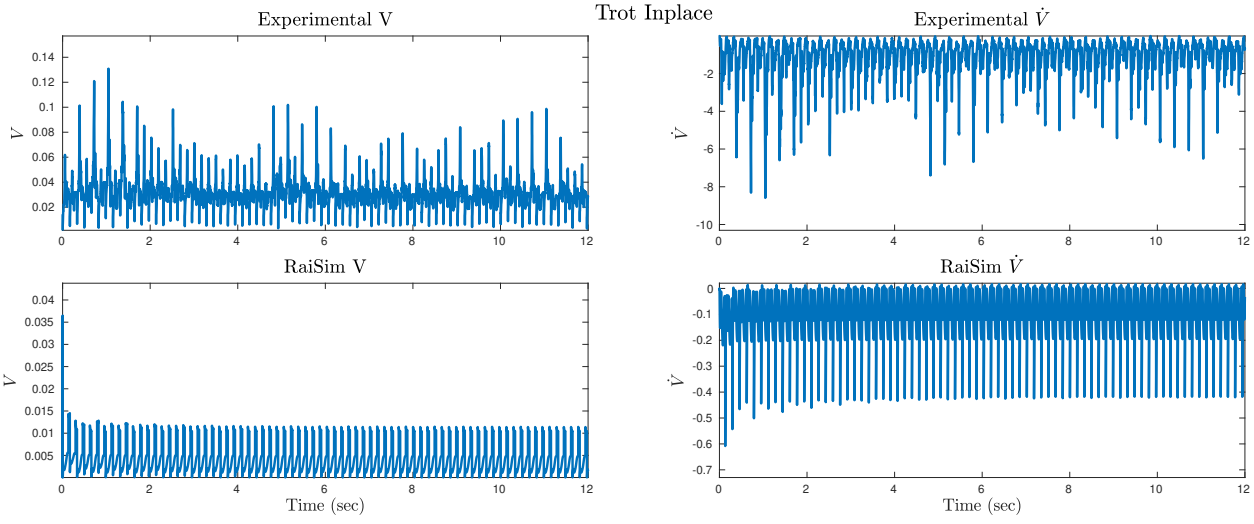

Figure 4.31: The simulated and experimental Lyapunov function corresponding to Fig. 4.29.

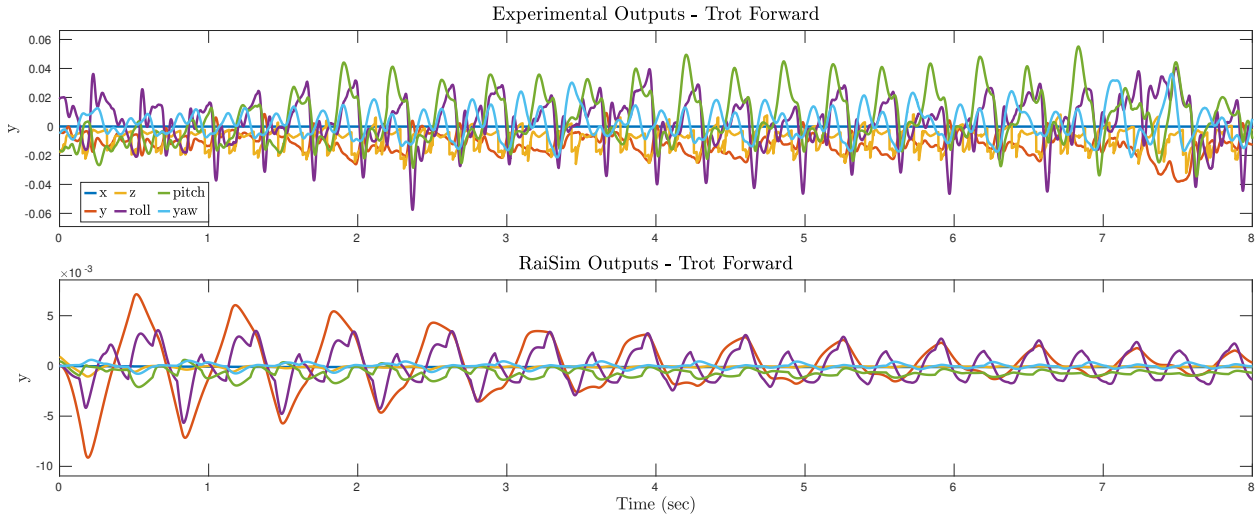

Figure 4.32: The simulated and experimental outputs (i.e., virtual constraints) for a forward trot with a command velocity of 0.1 (m/s).

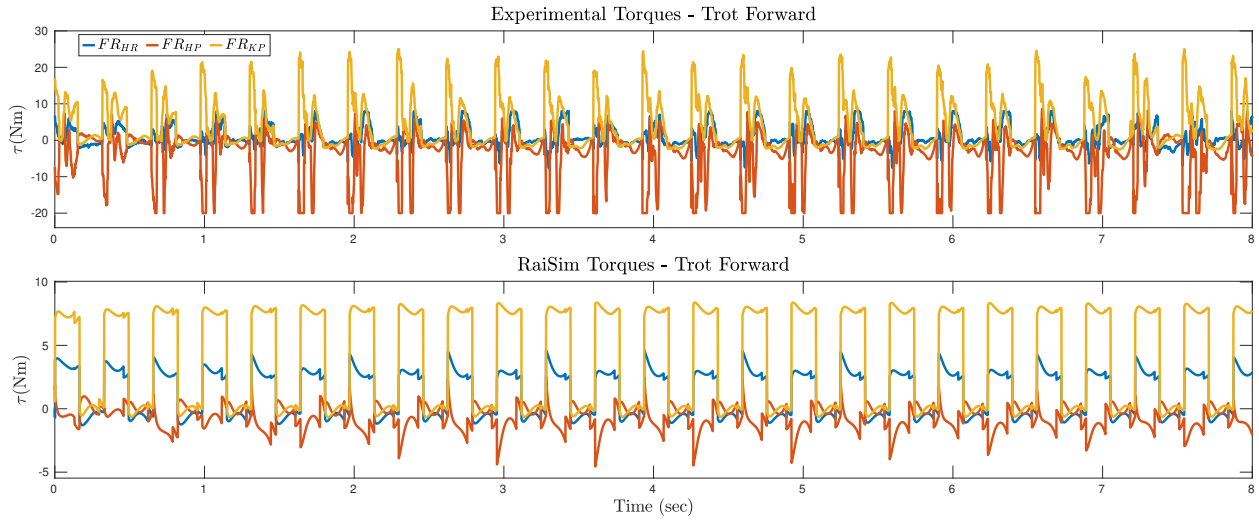

Figure 4.33: The simulated and experimental torques for the front right leg corresponding to Fig. 4.32.

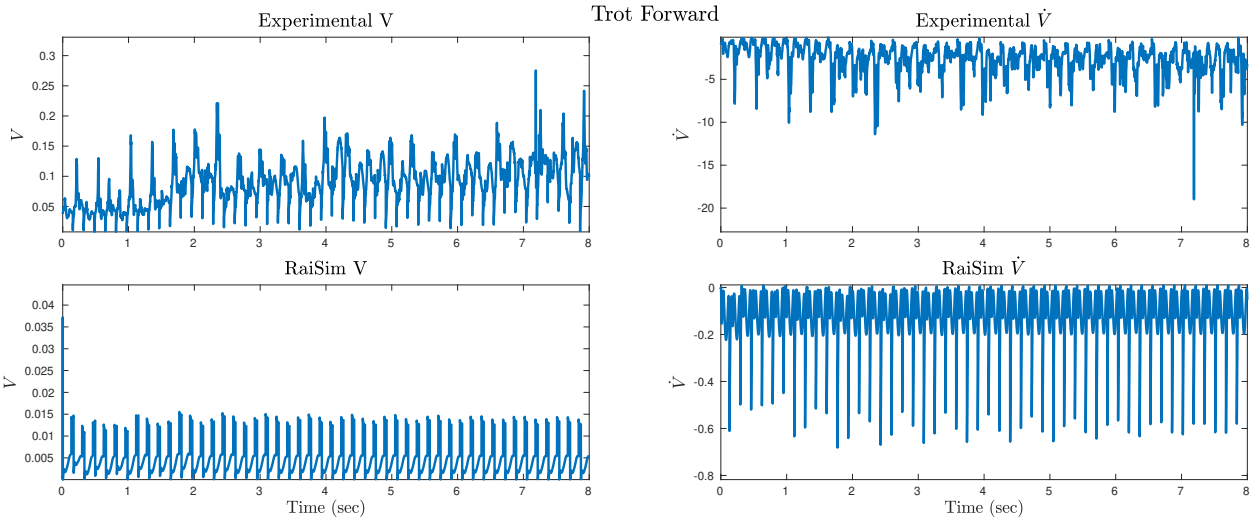

Figure 4.34: The simulated and experimental Lyapunov function corresponding to Fig. 4.32.

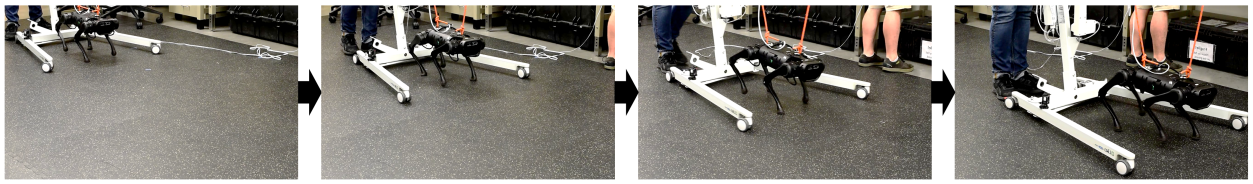

Figure 4.35: Snapshots of the experimental video for the forward trot corresponding to Fig. 4.32.

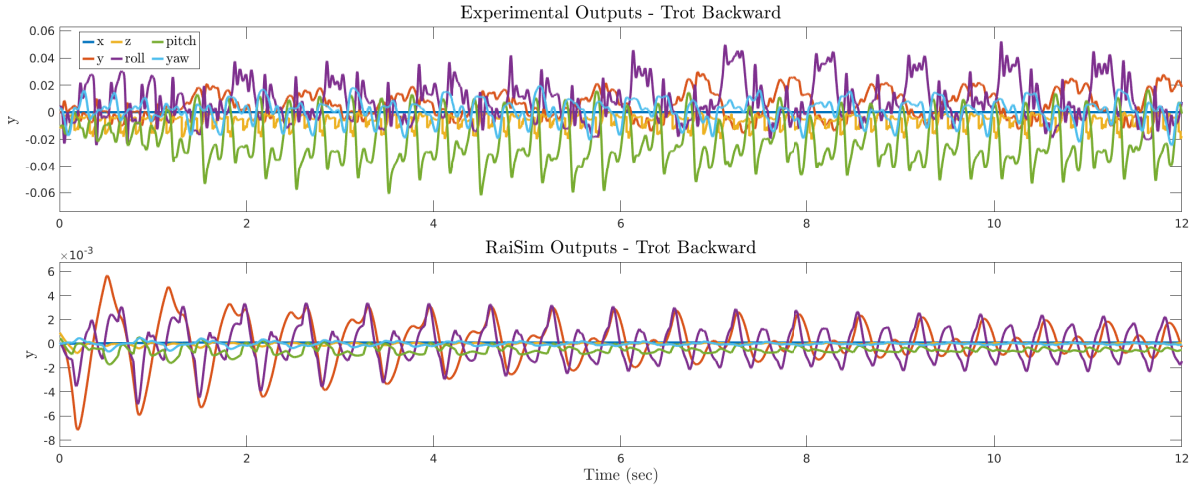

Figure 4.36: The simulated and experimental outputs (i.e., virtual constraints) for a backward trot with a command velocity of -0.1 (m/s).

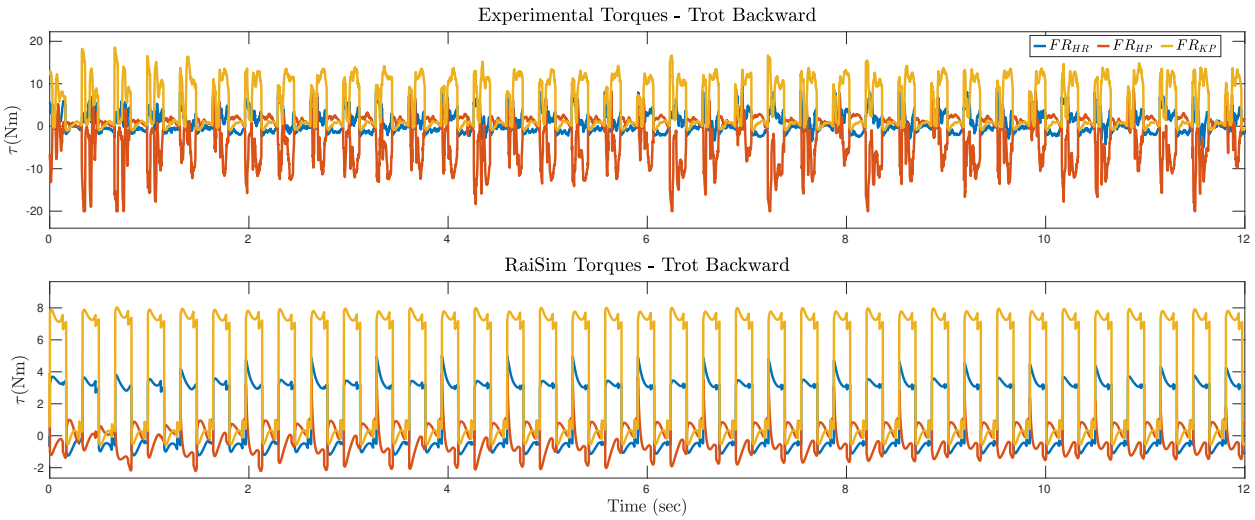

Figure 4.37: The simulated and experimental torques for the front right leg corresponding to Fig. 4.36.

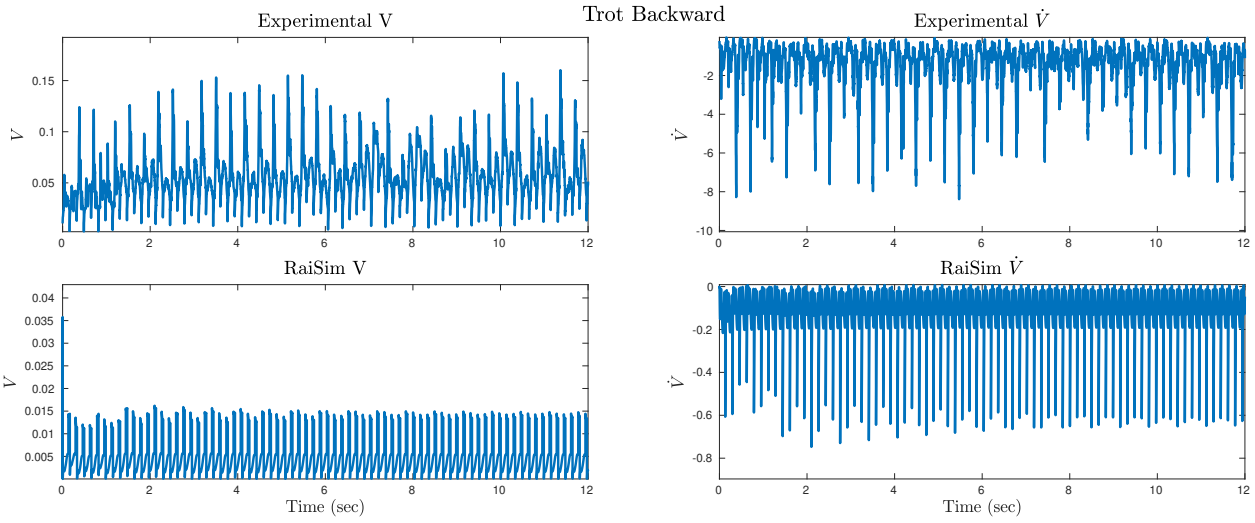

Figure 4.38: The simulated and experimental Lyapunov function corresponding to Fig. 4.36.

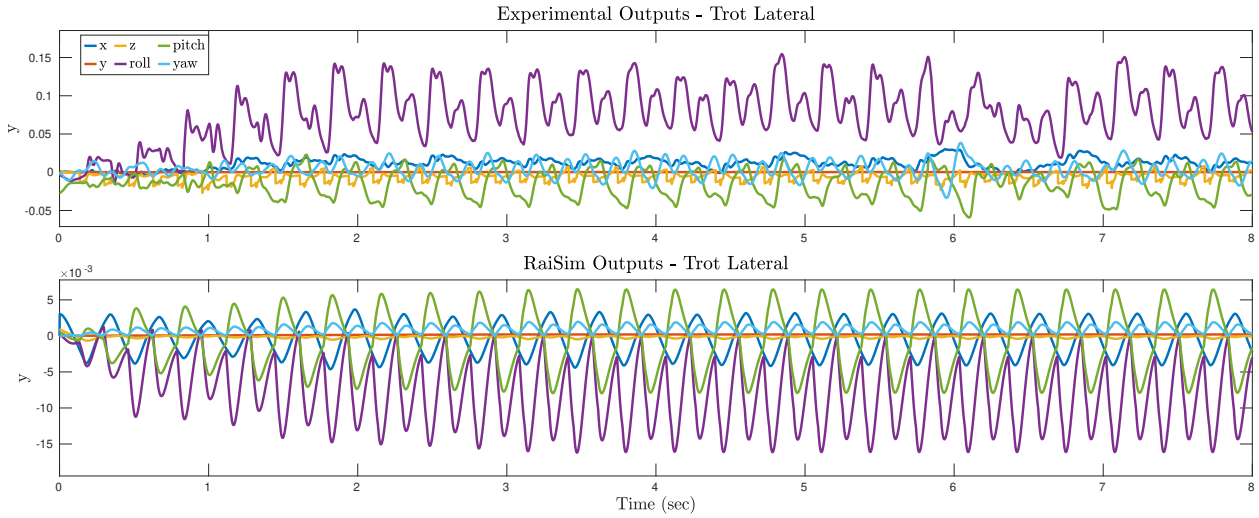

Figure 4.39: The simulated and experimental outputs (i.e., virtual constraints) for a backward trot with a command velocity of -0.1 (m/s).

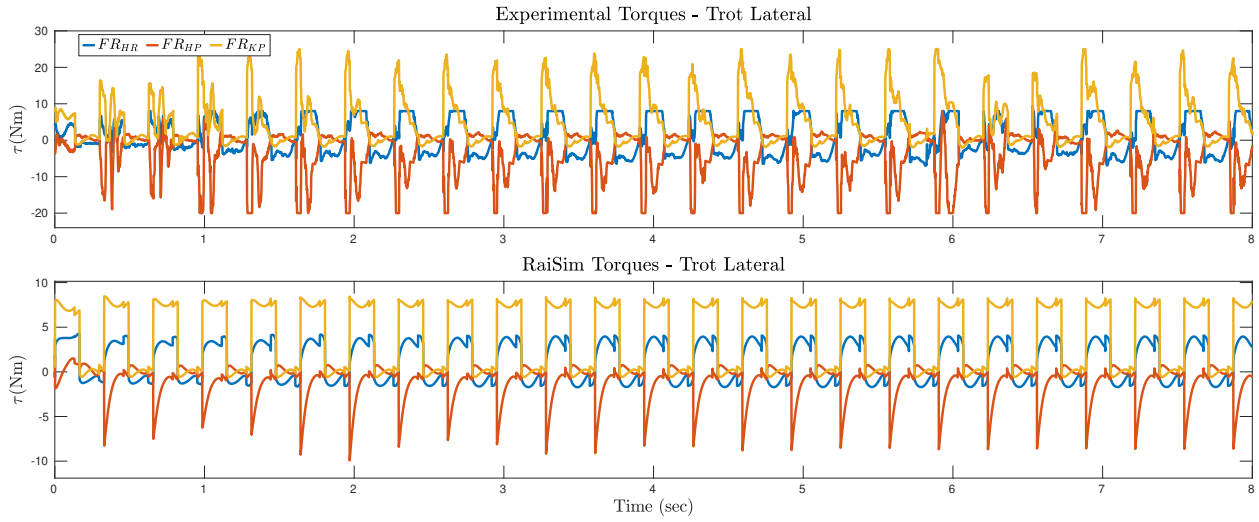

Figure 4.40: The simulated and experimental torques for the front right leg corresponding to Fig. 4.39.

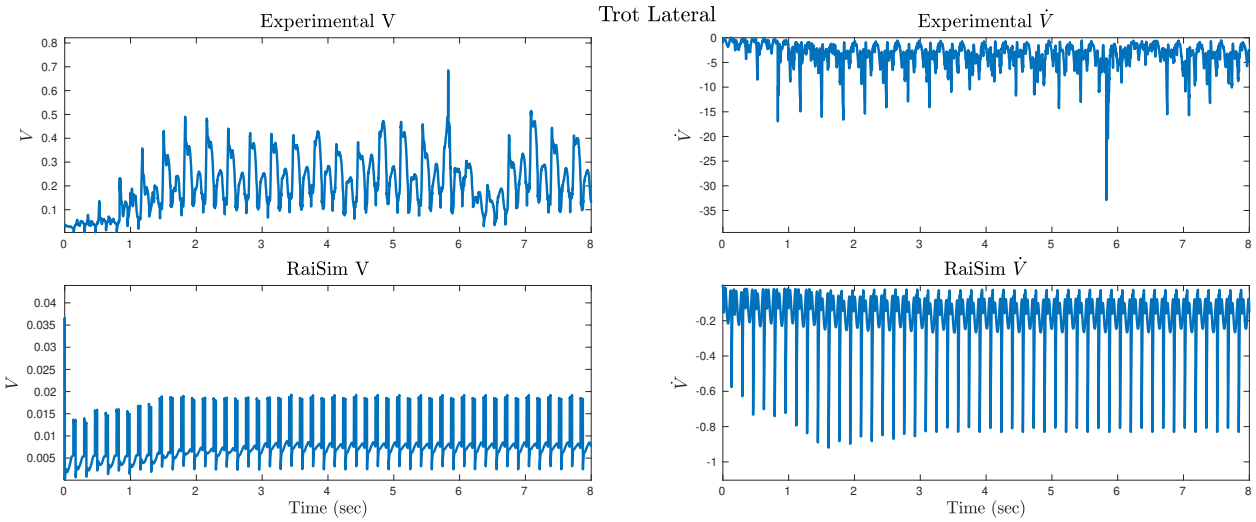

Figure 4.41: The simulated and experimental Lyapunov function corresponding to Fig. 4.39.

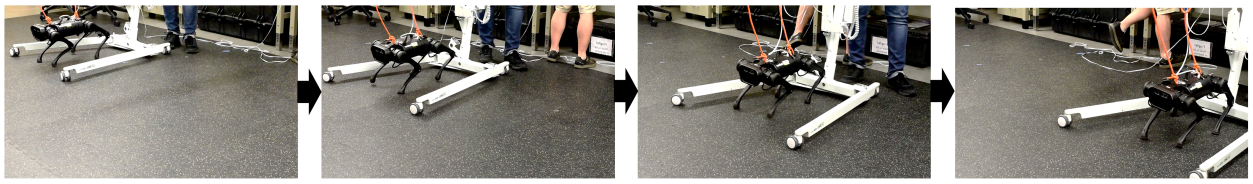

Figure 4.42: Snapshots of the experimental video for the lateral trot corresponding to Fig. 4.39.

Chapter 5

Conclusion and Future Work

5.1 Discussion

5.1.1 Quasi-Static Locomotion with Tails

The numerical studies and the full-order simulations of this work show significant improvements in rejecting both pulse-like and constant time-varying disturbances during quadrupedal locomotion when integrated with robotic tails under the proposed hierarchical control scheme in Chapter 3. The magnitude of increase in robustness depends largely on a variety of gait parameters (gait pattern, speed, direction, etc.) and the properties of the disturbance that is applied. The tail mechanism integrated with the proposed hierarchical control algorithm has been shown to allow the closed-loop system to withstand pulse-like disturbances up to 32.6% and 25.6% greater in magnitude during forward and diagonal trots when compared to a system without a tail. Furthermore, the tailed closed-loop system was able to withstand persistent perturbations up to 21.5% and 73% greater in magnitude during forward and diagonal trots, respectively.

Even with these relatively large increases in robustness to external forces, the tail does not produce highly aggressive motions during locomotion. Due to the nonlinearities of the tails shape dynamics, only a small angular deflection at the base of the yaw links is necessary to deflect the end effector of the tail greatly. Consequently, these small motions are able to

augment the stance dynamics enough to provide considerable assistance in rejecting external disturbances during locomotion. In addition, the locomotion considered here is quasi-static where it is reasonable to expect that the tail motions would take on trajectories that are also quasi-static. In more agile locomotion, specifically when quickly navigating rough terrain and turning at high speeds, the tail would likely produce much faster motions with larger deflections. Lastly, is is worth noting that the desired angle of the roll DOF of the tail remains essentially constant throughout the motions examined in simulation. In this work, the external disturbances were all created such that they are parallel to transverse plane. It follows that the optimal tail trajectories would also remain in the transverse plane, therefore allowing the majority of the energy expenditure for the tail to be contributing to rejecting the push disturbances. Therefore, when navigating flat ground with disturbances along the transverse plane, the yaw DOF of the tail plays a much greater role in disturbance attenuation.

The numerical results of the quasi-static locomotion in Chapter 3 have shown that the event-based framework coupled with the assumption that the roll, pitch, and yaw of the body are zero, along with the height of the COM being constant, leads to robust locomotion when maneuvering over flat terrain. In particular, under these conditions, successively linearizing the extended LIP model in an event-based manner at the beginning of each new continuous time domain is sufficient due to the slow changing nature of a quasi-static gait and the relatively small deflections at the base of the tail. It is noteworthy, however, that the methodology for the high-level path planner would need to be altered in order to provide robust locomotion along more complex environments wherein the terrain is unknown, unstructured, or otherwise difficult to traverse. This is the case for three reasons in particular, 1) in the current framework, the desired footstep locations are determined offline in the global coordinate frame in a heuristic manner, 2) moving up and down slopes would definitively

violate the assumptions pertaining to the height and orientation of the COM, and 3) the event-based linearization/planning may not be sufficient to adequately plan for locomotion across unstructured terrain considering higher planning rates are typically required when the environment is not well known.

5.1.2 Agile Dynamic Locomotion

As mentioned above, the tail provides very promising results for disturbance rejection during quasi-static locomotion. However, the primary end goal is to produce robust locomotion that is also agile, which is largely the reason for the exploration of different template models to produce agile locomotion. In particular, the use of the SRB model in conjunction with a higher update rate allows for much more dynamic high-level planning. The manner in which this template model is constructed, specifically the fact that additional forces being applied to the quadruped is a straightforward extension of this model, allows for the addition of a tail in a relatively simple manner. While this is not one of the concerns addressed in this work, one of the goals of this work is to create a controller which is amenable to adding a tail, or any other appendages coupled via an interaction force, in the future. Although the quadruped is able to walk without the high-level planner, and can track heuristic trajectories with very little tuning, more aggressive motions and locomotion along complex terrain become increasingly unstable without the addition of the planning level to provide better conditioned trajectories for the low-level controller to track. Specifically, the high-level planner contributes significantly to the medial-lateral stability of the quadruped during challenging scenarios and allows the quadruped to achieve stable gaits that the low-level controller alone cannot maintain. However, the scenarios during which the MPC with this template model play the largest role are when the system is being pushed to its limits, meaning that at low speeds over flat terrain, the low-level controller alone is sufficient to achieve

robust locomotion.

Even though the MPC with a higher update rate and more dynamic model provides many advantages over the LIP model, some of the robustness gained in the agile gait formulation comes from the footstep planning. Although it is heuristic, it provides excellent results and has been used on a large variety of legged locomotion platforms. Furthermore, the dynamic foot placement transfers to hardware in a much more robust manner due to errors in estimation when determining both COM location and foot location in the global coordinate frame. By virtue of the fact that the step locations are dynamic means that the foot steps can play a significant role in stabilizing locomotion and allows for better dynamic regulation of COM speed. Determining footstep locations dynamically also greatly improves robustness to various forms of disturbances and allows for dynamic push recovery, whereas the offline footstep locations creates a scenario in which, when the COM deviates too far from the aforementioned footstep locations, the system becomes unstable. This does not occur under the dynamic strategy as the quadruped step locations are relative to the body (in the $x-$ and $y-$directions), therefore allowing the quadruped to keep the feet under the body to recover from various disturbances.

Finally, the use of a CLF in the low-level controller allows for the QP to change the defect variables corresponding to the virtual constraints in a structured manner. In particular, assuming that the CLF inequality can be effectively met, the QP will allow the defect variables to change such that the output dynamics are Lyapunov stable at all times. Of course, this cannot necessarily be guaranteed at all times, which is why there is a slack variable on the CLF constraint. However, any time that this constraint is met, it guarantees that the system is converging exponentially to the output dynamics which is highly desirable for maintaining stable locomotion. Based on both the simulations and experiments, the CLF condition is generally met, and when it is not, $\max(\dot{V})$ has a tendency to be very small, and

does not typically remain positive for the majority of the stance domain.

The numerical results provided for the agile formulation show significant increases in the agility that may be achieved in a highly robust manner. The quadruped is able to quickly traverse challenging terrain, move at high speeds, and reject various external disturbances while remaining stable.

5.1.3 Experimental Validation

The experimental and simulated results shown in Section 4.5.2 show very good potential for the use of the proposed low-level controller. There are differences between the simulated and experimental results, most notably, tracking error, torques, and noise. Since the simulated scenarios do not consider sensor noise, friction, gear cogging torque, compliance at the foot, and many other factors, it is reasonable that the simulations do not align perfectly with experiments. However, the experimental data does have similar trends compared to the simulation data, and the torques are relatively close in magnitude between the two in most cases, particularly during pose control. It is worth noting that the largest issue left to overcome is swing leg tracking, which is believed to be the most pressing issue before better hardware results can be achieved. It is evident that the swing leg does not always reach the ground before the beginning of the next domain, leading to sub-optimal tracking performance. Due to the fact that foot placement plays a very large role in producing stable locomotion of legged robots, alternative approaches may be necessary to produce resilient swing leg tracking on hardware. Unfortunately, many estimators for the COM position and orientation rely heavily on contact assumptions. Therefore, the swing leg tracking is not only extremely important for tracking COM trajectories, but also for determining the current location and velocity of the COM. However, the preliminary results provided are

the first experimental validation of using a model-based CLF for quadrupedal locomotion. Though the controller has to undergo additional tuning, the fact that the quadruped is able to locomote in a stable manner under this choice of control is a considerable step forward in terms of the next generation of controllers for legged robots.

5.2 Summary of Contributions

The first major contribution of this thesis is the presentation of a hierarchical feedback control algorithm designed to effectively couple bio-inspired tails with legged robots, resulting in robust locomotion in the presence of external disturbances. In the proposed approach, the event-based high-level MPC is used to compute the optimal COM trajectory for the robot in addition to the tail trajectories by utilizing a novel reduced-order model, referred to as the extended LIP model. In this MPC formulation, the feasibility of the net GRF and the tail motion during the steering problem of the extended LIP model are considered. These trajectories are then passed down and tracked using a nonlinear low-level controller, based on virtual constraints and QP, in order to consider the full-order whole-body dynamics. The low-level QP is further restricted to ensure feasible GRF at each contacting leg end, as well as enforcing torque limits for all of the joints of the tail in addition to the quadruped. The results from this proposed approach were then evaluated through extensive simulations using an 18-DOF quadruped robot (Vision60) augmented with a 2-DOF serpentine tail. It is shown that, using this proposed control approach and the addition of the tail, produces locomotion that is more robust to external disturbances.

In the latter portion of this work, the aforementioned contributions are extended in order to take strides toward more agile locomotion. In particular, the extended LIP template model is replaced by a SRB-based template which lends itself more naturally to dynamic

motions. This template model is then linearized using VBL, resulting in a singularity free representation of the rotational dynamics for relatively small changes in rotation. A successively linearized MPC is then solved at 200Hz in order to provide forces which the low-level controller can then track such that the trajectory of the full-order system closely follows the trajectory created by the MPC. The low-level controller developed in Chapter 3.5 is also extended such that it includes an additional constraint, namely the CLF which allows for relaxations on the bounds of the defect variable used on the virtual constraints. The CLF provides some theoretical guarantees, but also aids in providing better conditioning of the problem for the QP such that faster and more reliable convergence can be achieved. The results from this secondary approach are also numerically validated over a variety of scenarios using a smaller 18-DOF quadrupedal robot, A1, and has shown significant improvements in robustness. These results, particularly those corresponding to the low-level controller augmented with a CLF, are also experimentally validated making this the first work in which an HZD-based controller without any approximation or optimized offline trajectory has been implemented on a quadrupedal hardware platform. Furthermore, this is the first work to consider and implement a model-based CLF on hardware for quadrupedal locomotion.

5.3 Future Work

There are many open avenues for future work, both in terms of standard quadrupedal locomotion, as well as locomotion of quadrupeds augmented with tails. In terms of the tails, one of the primary goals is to extend the use of tails to more agile locomotion, such as that presented in Chapter 4. This includes utilizing a CLF that considers the tail states and dynamics, and adding the tail to the reduced-order template model defined in Section 4.2.3 to take advantage of the dynamic nature of the model. Other avenues, such as exploring

other well established template models in order to achieve agile locomotion, such as the spring loaded inverted pendulum model, could also be explored. Finally, experimental validation of a quadruped augmented with a tail for robust disturbance rejection is of significant interest.

Considering extensions of the work of this thesis to general quadrupedal locomotion, one of the most important avenues of study will be exploring rough and varying terrain in both simulation and experimentation. In particular, in order for a quadruped to truly be capable enough for hands off autonomy, it should be able to adequately traverse hills, stairs, rough, and possibly loose terrain. In these areas there are a multitude of open questions ranging from control difficulties to accurate contact aided estimation techniques that can address issues such as foot slippage.

Lastly, there is a gap in the literature pertaining to the theoretical guarantees related to the path planners and low-level controllers, which provides additional avenues for continued research. This includes attempting to prove some of the theoretical guarantees regarding stability, as well as fundamental robustness analysis to quantify the theoretical robustness gained when using a nonlinear low-level controller. These analytical results would provide very good insight into the design of subsequent controllers and could greatly influence the direction of future research into robotic legged locomotion.

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] Hae-Won Park, Sangin Park, and Sangbae Kim. Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3d running of mit cheetah 2. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5163–5170. IEEE, 2015.

[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] ANYbotics. ANYmal Robot, https://www.anybotics.com/anymal-autonomous-legged-robot/, 2021.

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

[9] 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.

[10] M. Vukobratović, B. Borovac, and D. Surla. *Dynamics of Biped Locomotion*. Springer, 1990.

[11] A. Goswami. Postural stability of biped robots and the foot-rotation indicator (FRI) point. *The International Journal of Robotics Research*, 18(6):523–533, 1999.

[12] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of Honda humanoid robot. In *Robotics and Automation. Proceedings IEEE International Conference on*, volume 2, pages 1321–1326 vol.2, May 1998.

[13] M. Vulobratocić and B. Borovac. Zero-moment point — thirty five years of its life. *International Journal of Humanoid Robotics*, 01(01):157–173, 2004.

[14] H.-O. Lim, Y. Yamamoto, and A. Takanishi. Control to realize human-like walking of a biped humanoid robot. In *Systems, Man, and Cybernetics, IEEE International Conference on*, volume 5, pages 3271–3276 vol.5, 2000.

[15] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi. Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking. In *Robotics and Automation. Proceedings IEEE International Conference on*, volume 1, pages 368–374 vol.1, 1999.

[16] J.H. Park and K.D. Kim. Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. In *Robotics and Automation. Proceedings IEEE International Conference on*, volume 4, pages 3528–3533 vol.4, May 1998.

[17] K. Erbatur and O. Kurt. Natural ZMP trajectories for biped robot reference generation. *Industrial Electronics, IEEE Transactions on*, 56(3):835–845, March 2009.

[18] B.-J. Lee, D. Stonier, Y.-D. Kim, J.-K. Yoo, and J.-H. Kim. Modifiable walking pattern

of a humanoid robot by using allowable ZMP variation. *Robotics, IEEE Transactions on*, 24(4):917–925, Aug 2008.

[19] N. Motoi, M. Ikebe, and K. Ohnishi. Real-time gait planning for pushing motion of humanoid robot. *Industrial Informatics, IEEE Transactions on*, 3(2):154–163, May 2007.

[20] S. Kajita, T. Nagasaki, K. Kaneko, and H. Hirukawa. ZMP-based biped running control. *Robotics Automation Magazine, IEEE*, 14(2):63–72, June 2007.

[21] B. Ugurlu and A. Kawamura. ZMP-based online jumping pattern generation for a one-legged robot. *Industrial Electronics, IEEE Transactions on*, 57(5):1701–1709, May 2010.

[22] Tunc Akbas, S Emre Eskimez, Selim Ozel, O Kemal Adak, Kaan C Fidan, and Kemalettin Erbatur. Zero moment point based pace reference generation for quadruped robots via preview control. In *2012 12th IEEE International Workshop on Advanced Motion Control (AMC)*, pages 1–7. IEEE, 2012.

[23] Dong-Oh Kang, Yun-Jung Lee, Seung-Ha Lee, Yeh Sun Hong, and Zeungnam Bien. A study on an adaptive gait for a quadruped walking robot under external forces. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 2777–2782. IEEE, 1997.

[24] Takumi Kamioka, Hiroyuki Kaneko, Toru Takenaka, and Takahide Yoshiike. Simultaneous optimization of ZMP and footsteps based on the analytical solution of divergent component of motion. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1763–1770. IEEE, 2018.

[25] Gyunghoon Park, Jung Hoon Kim, Joonhee Jo, and Yonghwan Oh. Lyapunov-based

approach to reactive step generation for push recovery of biped robots via hybrid tracking control of dcm. *International Conference on Intelligent Robots and Systems*, 2020.

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

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

[28] 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.

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

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

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] 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.

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

[48] Sushant Veer, Rakesh, and Ioannis 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, 2019.

[49] 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, April 2018.

[50] 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.

[51] 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.

[52] 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 Berlin Heidelberg.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

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

[59] 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.

[60] 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.

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

[62] 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.

[63] 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.

[64] 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 *American Control Conference (ACC)*, pages 4588–4595, July 2019.

[65] 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.

[66] Kaveh Akbari Hamed, Jeeseop Kim, and Abhishek Pandala. Quadrupedal locomotion via event-based predictive control and QP-based virtual constraints. *IEEE Robotics and Automation Letters*, 5(3):4463–4470, 2020.

[67] K. A. 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 *2020 American Control Conference (ACC)*, pages 5314–5321, 2020.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

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

[79] 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.

[80] 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.

[81] 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.

[82] Ayonga Hereid and Aaron D. Ames. FROST: Fast robot optimization and simulation toolkit. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 719–726, Vancouver, BC, Canada, September 2017.

[83] K. Y. Chao, M. J. Powell, A. D. Ames, and P. Hur. Unification of locomotion pattern generation and control Lyapunov function-based quadratic programs. In *American Control Conference (ACC), 2016*, pages 3910–3915. IEEE, 2016.

[84] 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.

[85] 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.

[86] 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.

[87] 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.

[88] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture point: A step toward humanoid push recovery. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207, Dec 2006.

[89] 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.

[90] 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 vol.2, Sep. 2003.

[91] 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.

[92] 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.

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

[94] Yanran Ding, Abhishek Pandala, Chuanzheng Li, Young-Ha Shin, and Hae-Won Park. Representation-free model predictive control for dynamic motions in quadrupeds. *IEEE Transactions on Robotics*, pages 1–18, 2021.

[95] 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.

[96] 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.

[97] S. Fahmi, C. Mastalli, M. Focchi, and C. Semini. Passive whole-body control for quadruped robots: Experimental validation over challenging terrain. *IEEE Robotics and Automation Letters*, 4(3):2553–2560, July 2019.

[98] S. Kuindersma, F. Permenter, and R. Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *IEEE International Conference on Robotics and Automation*, pages 2589–2594, May 2014.

[99] Guan Horng Liu, Hou Yi Lin, Huai Yu Lin, Shao Tuan Chen, and Pei Chun Lin. A bio-inspired hopping kangaroo robot with an active tail. *Journal of Bionic Engineering*, 11(4):541–555, 2014.

[100] Steve W Heim, Mostafa Ajallooeian, Peter Eckert, Massimo Vespignani, and Auke Jan Ijspeert. On designing an active tail for legged robots: simplifying control via decoupling of control objectives. *Industrial Robot: An International Journal*, 43(3):338–346, 2016.

[101] Randall Briggs, Jongwoo Lee, Matt Haberland, and Sangbae Kim. Tails in biomimetic design: Analysis, simulation, and experiment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1473–1480. IEEE, 2012.

[102] Thomas Libby, Aaron M Johnson, Evan Chang-Siu, Robert J Full, and Daniel E Koditschek. Comparative design, scaling, and control of appendages for inertial reorientation. *IEEE Transactions on Robotics*, 32(6):1380–1398, 2016.

[103] Jianguo Zhao, Tianyu Zhao, Ning Xi, Fernando J Cintrón, Matt W Mutka, and Li Xiao. Controlling aerial maneuvering of a miniature jumping robot using its tail. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3802–3807. IEEE, 2013.

[104] Anna L Brill, Avik De, Aaron M Johnson, and Daniel E Koditschek. Tail-assisted rigid and compliant legged leaping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6304–6311. IEEE, 2015.

[105] Alan Mutka, Matko Orsag, and Zdenko Kovacic. Stabilizing a quadruped robot locomotion using a two degree of freedom tail. *2013 21st Mediterranean Conference on Control and Automation, MED 2013 - Conference Proceedings*, pages 1336–1342, 2013.

[106] Avik De and Daniel E Koditschek. Parallel composition of templates for tail-energized planar hopping. In *IEEE International Conference on Robotics and Automation*, pages 4562–4569. IEEE, 2015.

[107] Wael Saab, William S. Rone, Anil Kumar, and Pinhas Ben-Tzvi. Design and Integration of a Novel Spatial Articulated Robotic Tail. *IEEE/ASME Transactions on Mechatronics*, 24(2):434–446, 2019.

[108] William S. Rone, Wael Saab, and Pinhas Ben-Tzvi. Design, Modeling, and Optimization of the Universal-Spatial Robotic Tail. *ASME International Mechanical Engineering Congress and Exposition*, pages 1–10, 2019.

[109] Yujiong Liu, Jiamin Wang, and Pinhas Ben-Tzvi. A cable length invariant robotic tail using a circular shape universal joint mechanism. *Journal of Mechanisms and Robotics*, 11(5), 2019.

[110] William S Rone and Pinhas Ben-Tzvi. Continuum robotic tail loading analysis for mobile robot stabilization and maneuvering. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 46360, page V05AT08A009. American Society of Mechanical Engineers, 2014.

[111] Chunlei Rui, Ilya V Kolmanovsky, and N Harris Mcclamroch. Nonlinear Attitude and Shape Control of Spacecraft with Articulated Appendages and Reaction Wheels. *IEEE Transactions on Automatic Control*, 45(8):1455–1469, 2000.

[112] G. Wenger, A. De, and D. E. Koditschek. Frontal plane stabilization and hopping with a 2dof tail. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 567–573, 2016.

[113] Thomas Libby, Talia Y Moore, Evan Chang-Siu, Deborah Li, Daniel J Cohen, Ardian Jusufi, and Robert J Full. Tail-assisted pitch control in lizards, robots and dinosaurs. *Nature*, pages 2–7, 2012.

[114] Wael Saab and Pinhas Ben-Tzvi. Design and analysis of a discrete modular serpentine robotic tail for improved performance of mobile robots. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 50152, page V05AT07A061. American Society of Mechanical Engineers, 2016.

[115] Joseph Norby, Jun Yang Li, Cameron Selby, Amir Patel, and Aaron M Johnson. Enabling dynamic behaviors with aerodynamic drag in lightweight tails. *IEEE Transactions on Robotics*, 2021.

[116] Amir Patel, Edward Boje, Callen Fisher, Leeann Louis, and Emily Lane. Quasi-steady state aerodynamics of the cheetah tail. *Biology open*, 5(8):1072–1076, 2016.

[117] Yuuki Mishima and Ryuta Ozawa. Design of a robotic finger using series gear chain mechanisms. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1(IROS):2898–2903, 2014.

[118] Yasuyuki Hirano, Kensaku Akiyama, and Ryuta Ozawa. Design of Low-Cost and Easy-Assemblable Robotic Hands with Stiff and Elastic Gear Trains. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 864–870, 2016.

[119] A Jusufi, D T Kawano, T Libby, and Robert J Full. Righting and turning in mid-air using appendage inertia : reptile tails , analytical models and bio-inspired robots. *Bioinspiration and Biomimetics*, 2010.

[120] E Passerello and R L Hustons. Human Attitude Control. *Journal of Biomechanics*, 4:95–102, 1971.

[121] T. R. Kane and M. P. Scher. Human self-rotation by means of limb movements. *Journal of Biomechanics*, 3(1):39–49, 1970.

[122] David Attenborough. The life of Mammals: the Complete Series - Episode 5: meat Eaters. BBC Documentary, 2002 - 2003.

[123] Amir Patel and M Braae. An actuated tail increases rapid acceleration manoeuvres in quadruped robots. In *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, pages 69–76. Springer, 2015.

[124] Konstantinos Machairas and Evangelos Papadopoulos. On quadruped attitude dynamics and control using reaction wheels and tails. In *European Control Conference*, pages 753–758, 2015.

[125] Wael Saab and Pinhas Ben-Tzvi. Maneuverability and heading control of a quadruped robot. In *ASME Dynamic Systems and Control Conference*, pages 1–7, 2017.

[126] Carlos Casarez, Ivan Penskiy, and Sarah Bergbreiter. Using an inertial tail for rapid turns on a miniature legged robot. In *2013 IEEE International Conference on Robotics and Automation*, pages 5469–5474. IEEE, 2013.

[127] N. J. Kohut, A. O. Pullin, D. W. Haldane, D. Zarrouk, and R. S. Fearing. Precise dynamic turning of a 10 cm legged robot on a low friction surface using a tail. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3299–3306, 2013.

[128] A. O. Pullin, N. J. Kohut, D. Zarrouk, and R. S. Fearing. Dynamic turning of 13 cm robot comparing tail and differential drive. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5086–5093, 2012.

[129] Evan Chang-siu, Thomas Libby, Masayoshi Tomizuka, and Robert J Full. A Lizard-Inspired Active Tail Enables Rapid Maneuvers and Dynamic Stabilization in a Terrestrial Robot. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1887–1894, 2011.

[130] Amir Patel and M. Braae. Rapid turning at high-speed: Inspirations from the cheetah's tail. *IEEE International Conference on Intelligent Robots and Systems*, pages 5506–5511, 2013.

[131] Xiuli Zhang, Jiaqing Gong, and Yanan Yao. Effects of head and tail as swinging appendages on the dynamic walking performance of a quadruped robot. *Robotica*, 34(12):2878–2891, 2016.

[132] Martin L. Felis. RBDL: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots*, pages 1–17, 2016.

[133] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference*, pages 3071–3076, July 2013.

[134] 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.

[135] C.C. De Wit, H. Olsson, K.J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, Mar 1995.

[136] Marc H Raibert, H Benjamin Brown Jr, and Michael Chepponis. Experiments in balance with a 3d one-legged hopping machine. *The International Journal of Robotics Research*, 3(2):75–92, 1984.

[137] Wouter J. Wolfslag, Christopher McGreavy, Guiyang Xin, Carlo Tiseo, Sethu Vijayakumar, and Zhibin Li. Optimisation of body-ground contact for augmenting the whole-body loco-manipulation of quadruped robots. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3694–3701, 2020.

[138] Romeo Orsolino, Michele Focchi, Carlos Mastalli, Hongkai Dai, Darwin G Caldwell, and Claudio Semini. Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots. *IEEE Robotics and Automation Letters*, 3(4):3363–3370, 2018.

[139] Yanran Ding, Chuanzheng Li, and Hae-Won Park. Kinodynamic motion planning for multi-legged robot jumping via mixed-integer convex program. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3998–4005, 2020.

[140] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. Stable manifolds of saddle equilibria for pendulum dynamics on s 2 and so (3). In *2011 50th IEEE conference on decision and control and European control conference*, pages 3915–3921. IEEE, 2011.

[141] 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.

[142] Peter A Gorry. General least-squares smoothing and differentiation by the convolution (savitzky-golay) method. *Analytical Chemistry*, 62(6):570–573, 1990.