Efficient Modeling of Multi-legged Locomotion with Slipping

by

Ziyou Wu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Robotics)
in the University of Michigan
2024

Doctoral Committee:

Associate Professor Shai Revzen, Chair
Assistant Professor Maani Ghaffari Jadidi
Associate Professor Robert D. Gregg
Assistant Professor Talia Yuki Moore

Ziyou Wu
wuziyou@umich.edu
ORCID iD: 0000-0002-4543-6815

To my family

ACKNOWLEDGEMENTS

I express my deepest appreciation to my advisor, Professor Shai Revzen, for his mentorship since I was an undergraduate student. His breadth of knowledge sparked my interest in studying robotics, statistics, math, and biomechanics. I thank him for his crucial insights into this dissertation. He helped me understand science, philosophy, and the scientific value of research. His passion for interesting problems has always inspired me. His guidance over the years gradually shaped me into an independent researcher.

I also thank all the lab members. Dan Zhao was a senior graduate student when I joined the lab and was my mentor on my first project in the lab. I thank him for teaching me how to build BigAnt, helping me with debugging the inchworm model, and discussing the friction models. I thank Brian Bittner and George Council for many interesting conversations when I first joined the lab. I thank Marion Anderson for helping me out with electronics, soldering, and programming problems. I thank Tianle Wu for his help in collecting robot data, and thank Shuyu Zhan for his help with GPU coding. I thank Leo Prior, Nicholas Pagliocca, Yishun Zhou, and Kayla Hurst for joining me in taking care of cockroaches. It was a pleasure to work together with Daniel Kenneday, Erick Vega, Arun Bishop, Jin Suh, Tyson Lin, Advait Deshpande, Hao Luo, Haotian Li, Vikram Sachdeva, Dan Altman, and all other lab mates.

I would like to thank Liz Olson, Cynthia Lin, and Alia Gilbert for initiating the Women in Robotics and Engineering (WIRE+) community. This community provided much support and fun activities to connect women in the field. I am also grateful for Michigan's student-run Legged Locomotion Seminar, organized by Grant Gibson and Eva Mungai. I have learned a lot from it. Many thanks to the Group-X community for the fitness classes that motivated me to move.

Finally, I could not have undertaken this journey without the support of family and friends. To my parents, I thank your unconditional love, support, and encouragement leading me to the place where I am. The weekly family meeting together with my cousin Shangxing and my little niece Anya always brought me joy and relaxation. I thank my friend Mengqiu Teng for always being there. Our adventurous trips were a cherished memory. To my partner Ruowen, I am lucky to have your support, and your faith in my abilities more than I do myself. Life is so much more colorful with you by my side.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

Multi-legged robots with six or more legs have designs with great stability and maneuverability using a low number of actuators. With more legs in contact with the ground, it becomes harder and unnecessary to ensure that they all move in compatible ways to ensure non-slip contact, and slipping becomes inevitable in these robots. Modeling multi-legged motion with slipping and producing reliable predictions of body velocity is challenging and computationally expensive. In this work, we investigated an algorithm that allows us to efficiently model body velocity and ground contact forces given robot shape and shape changing velocity. This algorithm relies on previous experimental observations showing that even while slipping, multi-legged robots are principally kinematic, where body velocity could be computed through a linear local connection from the shape changing velocity. This tool scaled well with an increasing number of legs without losing accuracy. We applied our algorithm towards modeling multiple robots with different morphology and different gaits, together with predicting 3-dimensional ground contact forces. Further extension of this model enabled its usage on stairs or slopes. The simplicity of the model led to easy parallelization and motion planning of the robot's behaviors on GPU. Analysis of the modeling residual term led to a simple and data-efficient sim-to-real transfer to unmodeled robot dynamics.

CHAPTER 1

Introduction

1.1 Motivation and background

Whether it is animals navigating their natural habitats or robots exploring unknown terrain, the ability to move through space is essential for both biological and artificial entities. Wheels have been developed for efficient land transportation for thousands of years in human civilizations, yet have rarely evolved as a means of terrestrial locomotion. Although wheel-based transportation is energy efficient on flat surfaces, it soon loses its advantage in uneven, muddy, or sandy terrain [1]. On the other hand, terrestrial animals have evolved legs as a means of locomotion on land, allowing them to navigate and interact with their environment. Legs provide support, stability, propulsion, and maneuverability, enabling animals to move efficiently across various terrains, whether it be crawling, walking, running, or jumping.

Among all legged animals, arthropods, usually with six or more legs are incredibly successful and abundant, with over a million described species, making up the largest phylum in the animal kingdom. Arthropods are highly adaptable to various environments, ranging from terrestrial to freshwater and marine habitats. Their exoskeleton provides protection and support, allowing them to inhabit diverse ecological niches. Such "multi-legged" systems – a term we refrain from using for quadrupeds and bipeds – have extra robustness over complex terrain [2]. They can achieve stability more easily with three or more points of contact and tolerate more slipping. Arthropods not only exist in decimeter or smaller scale, as fossil evidence shows there are giant myriapods (Arthropleura, 2 meters long, 0.5 meters wide, 50 kilograms weight) from the early Carboniferous to Early Permian [3]. The reasons they are much larger than modern arthropods partially lie in the higher oxygen concentration and lack of large terrestrial vertebrates. Since robots do not depend on respiration for power, insect-inspired robots should not have size limitations as biological insects do.

Much of the work on meter-sized multi-legged robots has focused on hexapedal robots, starting with [4, 5]. Hexapods are appealing because they can have a tripod of supporting legs while moving another tripod into place for support on the next step. This static stability allows for the possibility of easier control, and usable motion even when the robot is slipping. Significant prior work was done on the hexapedal robots of the RHex family [6, 7], many variations of which have been built over the past 20 years. The BigAnt robot is a low-cost fast-fabricated hexapedal robot with 1 degree of freedom (DoF) four-bar linkage legs [8]. Additional families of multi-legged robots include the RoACH robots of [9], the Sprawl robots of [10], various multi-legged robots that used "whegs" from [11] and e.g.[12], and several studies of multi-legged robots with larger numbers of legs such as [13, 14]. Bipedal and quadrupedal robots usually use articulated legs with many DoF, i.e. Atlas (6 DoF per leg), Spot (3 DoF per leg), whereas multi-legged robots usually use low-DoF legs with an average of 1 or even less DoF per leg [8]. This substantially lowers the cost of multi-legged robots, since a quadrupedal robot like Spot uses 12 motors, whereas Rhex or BigAnt only uses 6. Although multi-legged robots might not be as agile as quadrupedal or bipedal robots in executing dynamic behaviors, such as jumping or back flipping, they have the same capability of traversing rough terrain. This makes them a great candidate as a transportation platform for exploration, construction, search and rescue, and disaster relief.

Thanks to having many simultaneous points of contact rather than the single point of contact that bipeds often have, multi-legged animals can tolerate significant slipping on the ground. As an example of that, previous research shows that the feet (tarsi) of *Blaberus discoidalis* cockroaches running on a paper surface (Elmer's 900803 Foam Board, Elmer's Products Inc., Atlanta GA.) spend about 20% of their entire motion in the lab frame slipping backward while in contact with the ground [15]. It quickly becomes clear to us that as more and more legs are in contact with the ground it becomes harder to ensure that they all move in compatible ways to ensure non-slip contact, and slipping becomes inevitable in our robots. As the modeling complexity increases combinatorially with the number of legs when considering multiple contacts and slipping, a natural question to ask is: how to efficiently model multiple sliding contacts?

Template models help researchers understand the mechanisms and strategies in legged locomotion by reducing the complexity [16]. Templates and anchors were first developed to study the rhythmic movements of biological organisms, where a template uses a smaller number of variables to describe the behavior of interest, while an anchor turns it into a more biologically realistic model. For example, spring-loaded inverted pendulum (SLIP) model [17] and lateral leg spring (LLS) model [18, 19] are widely known template models for studying vertical and horizontal motion respectively. The template models facilitate many studies to design, optimize, and control for legged robots [6, 20, 21]. However, the template models usually ignore slipping, and the interplay between the contacting legs.

On the other hand, simulators or physics engines, such as PyBullet [22], MuJoCo[23] can take in the robot full body design and integrate robot dynamics together with contacts. These models usually involve a large number of parameters that need to be fine-tuned before transferring to the physical robots. Recent researches on reinforcement learning using such simulators enable quadrupedal robots to improve the simulation accuracy while learning a robust policy [24, 25]. These learning methods are quite successful in deploying real-world applications, yet they are computationally expensive and require billions of iterations with many hours of training. When moving towards multi-legged robots with slipping, the computation time of evaluating each simulation step would probably increase linearly or even combinatorially with more legs.

As the number of legs increases and the relative size of each leg becomes smaller, the locomotion of a multi-legged robot converges to that of a slithering snake robot. Slithering can be understood using geometric mechanics, and is "principally kinematic" in the sense of admitting equations of motion that do not require momentum or velocity states. Instead, the equations of motion can be expressed as a "local connection": a linear relationship between the rate of shape change $\dot{r}$ and the body velocity v_b: $v_b = A(r)\dot{r}$ (see e.g. [26, 27]). It is natural to ask how this transition to momentum-free equations of motion occurs with increasing numbers of legs, and whether we can obtain a model for multi-legged locomotion without momentum.

Somewhat to our surprise, we found that three slipping point contacts represent the minimal number required to remove momentum and describe motion using local connections. This is because, with this number of contacts, all degrees of freedom are either rigidly constrained or subject to sliding friction. The work in [28] presented a data-driven method to efficiently construct this local connection model for systems with no momentum. The work in [29] extended the no-momentum case to the quickly decaying momentum case which appears in high friction sliding contact. Furthermore, [30] showed that such local connection models are effective for multi-legged animals and robots with and without slipping, at the predicted level of three or more contacts. The simple structure makes modeling the body motion of multi-legged robots much easier, as despite the complexity of additional contacts and a combinatorial number of contact states, the resulting simplification, with momentum being ignorable, more than compensates.

In this thesis, we approach modeling multi-legged locomotion by constructing a simple connection model which accurately estimates the body velocities given the body shape movements, and also captures the internal loading and slipping among all legs. We hope this work could demonstrate to the community that the body motion of multi-legged robots – typically hexapods and octopods – is much easier to model than is often assumed. Even if each leg ends in a multi-toed foot, leading to a total of a few dozen possible contact points, the simulation models remain eminently tractable. This simple and efficient model can be easily extended to uneven terrain, and allows multi-legged robots to do motion planning by searching through their entire configuration space.

1.2 Contributions and overview

The contributions of this thesis are to:

1. Present a simple inchworm crawling model and show a linear relationship between averaged body velocity and propulsion force under Coulomb friction slipping,

2. Present a fast algorithm for modeling multi-legged systems, that accurately estimates the body velocities given the body shape movements, and provides accurate foot contact force estimates under slipping,

3. Present an experimental setup and calibration method for a hexapod with force/torque sensors measuring ground contact forces of each foot and thereby validate the predictions of our model,

4. Provide an open source dataset of the kinematic and force/torque measurements on BigAnt robot in [31],

5. Extend the proposed model to uneven terrain, transfer the model to robots with unmodeled dynamics, and apply it to motion planning.

1.2.1 Chapter 2: Coulomb friction crawling yields linear force-velocity profile

We used an over-simplified inchworm crawling model to analyze the relationship between averaged body velocity and the legged propulsion with slipping under Coulomb friction. We found the averaged velocity versus input force in Coulomb friction dominated motion is a linear relationship, at least for the parameter range governing small robots and animals (several kilograms with propulsion forces generating up to $2g$ acceleration). This discovery was quite surprising because conventional wisdom would have it that moving mechanical systems that dissipate energy by Coulomb friction have no relationship between force and average speed. Our numerical model, amenable to the exact solution, exhibited nearly linear ($R^2 > 0.96$) relationships between actuator force and average speed over its entire range of parameters in both motion regimes it supported. This suggested that the interactions inherent in multi-legged locomotion may lead to governing equations more reminiscent of viscous friction than would be immediately obvious.

1.2.2 Chapter 3: efficient multi-legged locomotion model

We presented a fast algorithm for modeling multi-legged systems, that accurately estimated the body velocities given the body shape movements, and provided accurate foot contact force estimates.

Based upon the previous work in chapter 4 of [8], in this work, we further improved the spring-supported model by allowing pitch and roll to better model the turning behavior, and allowed the friction to have an anisotropic direction. We validated this model against several multi-legged robot datasets, including one with force-torque sensors measuring the ground contact forces of each foot. We demonstrated that it scales favorably with the number of legs, which only doubles the computation time when the leg number increases from 3 to 50.

1.2.3 Chapter 4: experimental setup and calibration of hexapod BigAnt with force/torque sensors measuring individual ground contact wrenches

We presented an experiment setup that simultaneously measured the ground reaction force on each individual foot of a multi-legged robot with low DoF legs, BigAnt. We installed each leg with one 6-DoF force/torque transducer at the hip of the hexapod, enabling us to measure the wrench (forces and torque) interacting between the leg and robot chassis, while still maintaining the lightweight, fast-swinging nature of the legs. We presented an in-situ calibration procedure enabling simultaneous measurement of all 6-dimensional foot contact wrenches. Our calibration, without disassembling the setup, was able to calibrate (1) constant transducer offset, (2) phase-dependent leg gravity offset, (3) frame transformation error between the transducer frame and robot frame, and (4) unknown robot center of mass location. After calibration, the RSME was reduced by 63% for forces and 90% for torques in the residuals of the robot standing in different poses, compared with naive constant offset removal.

1.2.4 Chapter 5: model extensions and applications

In this chapter, we show some preliminary results for future work extending and applying this model. We presented the initial results of extending the model presented in Chapter 3 to uneven terrain, doing sim-to-real transfer, and applying it to motion planning. We relaxed the model assumption of a flat surface, and allowed the robot pitch and roll to vary around non-zero constant angles. This allowed the model to apply to slopes and staircases. We used a data-driven connection residual term to capture the unmodeled dynamics, and improved the motion prediction on side velocities for Multipod with leaf-spring legs using undulatory gait. We further implemented this model on GPU to plan the robot's gait with a large number of possible motions computed in parallel. We showed that simple brute-force motion planning can generate gaits that are not intuitive to design on a multi-legged robot with low-DoF legs.

CHAPTER 2

Inchworm Crawling Model

This chapter closely reproduces our paper [32].

2.1 Motivation

Conventional wisdom often believed that mechanical systems employing Coulomb friction for energy dissipation exhibit no correlation between force and average speed. The work done by friction is the product of the frictional force and the displacement over which the force acts. If propulsion forces exceed friction, the net work is positive and the system accumulates kinetic energy without bound. This is true for wheeled vehicles; when ignoring energy limit and other drag or resistance, a car can keep accelerating. On the other hand, legged animals use their limbs to move, through intermittent contacts with the substrate. The motion of the legs is restricted to a finite length range. It is not immediately clear what is the governing relationship between friction, actuation, and finite body size.

In this chapter, we seek to use a minimalistic model to study how to model legged systems with slipping under Coulomb friction. Rather than confounding the effects of slipping with those of intermittent contact, we chose to model the uniaxial motion of a two-mass system in continuous sliding contact. Our model was inspired by the shuffling gait people sometimes employ on ice or other slippery surfaces, whereby they shift their weight over one leg, slide the other leg to a new position, and repeat, alternating the weight-bearing leg. Alternatively, this can be thought of as an inchworm gait on a slippery surface. A similar two-mass model without mass shifting was used in [33] to show locomotion is possible when friction forces on two bodies and the contraction-extension phases are both asymmetric.

Our model, scaled to parameters representative of single kilogram robots and animals, exhibits nearly linear ($R^2 > 0.96$) relationships between actuator force and average speed over its entire range of parameters, and in both motion regimes, it supports. This suggests that even when frictional contacts are well modeled by the classical Coulomb friction model [34, 35], the average relationship between actuation force and body velocity can look remarkably similar to viscous friction [36]. We demonstrate this result with an overly simplistic model of locomotion with slipping, which is simple enough to allow us to analyze its solutions exhaustively. We also compare this model with a slightly more elaborate and physically plausible model, the results of which we only study numerically.

We begin by describing and motivating our sliding locomotion model in §2.2. In §2.3 we present the results of simulating the model over a range of parameters surrounding those applicable to small robots and animals. In §2.4 we solve for average force and velocity in closed form. The instantaneous mass-swapping of §2.2 is somewhat physically implausible, and so in §2.5 we present the results from a more physical model of mass swapping, which we analyze numerically. We follow by discussing these results in §2.6, where we hazard some guesses as to the cause and broader implications of having a force-velocity rather than a force-acceleration relationship.

2.2 Sliding Locomotion Model

2.2.1 Mechanical Model

Consider two point masses with an actuator of finite length between them (see figure 2.1) operating at constant force pushing them apart. Each of these masses represents a single leg, with the magnitude of the mass representing the loading on that leg. When the distance between two masses reaches a predetermined maximum, a perfectly plastic collision occurs. The positions of the two masses are swapped, representing a reversal of the leg loading, and the force flips sign to pull the masses together. Once they reach a minimal distance, they collide a plastic collision again, swap

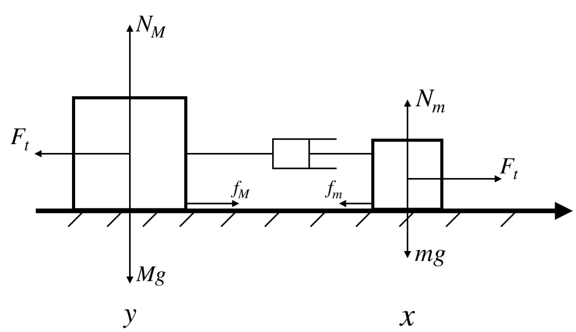

Figure 2.1: Schematic representation of the mechanical model. M and m are the larger and smaller point masses respectively, with y and x their positions. The friction forces acting on the point masses are f_m, f_M. F_t is actuator force, which is positive when pushing the masses apart, and negative when pulling them together. g is gravitational acceleration.

positions, the direction of force switches, and the cycle repeats. (A discussion of a more physical model of mass-swapping can be found in §2.5.)

The equation of motion of the system from Newton's law is:

$$m\ddot{x} = F_t \operatorname{sgn}(x - y) - f_m \tag{2.1}$$

$$M\ddot{y} = -F_t \operatorname{sgn}(x - y) - f_M \tag{2.2}$$

We assume the Coulomb friction model for the contact forces [37]. In the Coulomb friction model, the friction force is proportional to the normal load when velocity is not 0. When velocity is 0, the friction force can take any value between 0 and a static friction force proportional to the normal load.

This can be modeled using equations below:

$$f = \begin{cases} \mu N \operatorname{sgn}(v) & \text{slipping (dynamic friction)} \\ F_e & \text{non-slip; (static friction); } |F_e| < \mu_s N; \end{cases} \tag{2.3}$$

where μ is dynamic friction coefficient, μ_s is static friction coefficient, F_e is external force, and N is normal force. Both masses are taken as point mass with state vector $[x, \dot{x}]^T$ and $[y, \dot{y}]^T$ respectively. The distance between the masses is bounded above and below by L_{max} and L_{min} respectively. When either distance limit is reached, it is enforced by a perfectly plastic collision which renders equal the velocity of both masses:

$$\dot{x}^+ = \dot{y}^+ = \frac{m\dot{x}^- + M\dot{y}^-}{m + M} \tag{2.4}$$

Immediately after the masses collide, they instantaneously swap positions, i.e. the pre-collision state $[x, \dot{x}^-, y, \dot{y}^-]^\mathsf{T}$ gives rise to a post-collision state $[y, \dot{y}^+, x, \dot{x}^+]^\mathsf{T}$

This last model assumption factors out the details of exactly how the moving system redistributes load over its legs, and merely captures the notion that the leg load alternates between the two possible configurations.

2.2.2 Dynamic Analysis

Holding actuation force at constant magnitude, alternately extending and contracting the actuator through its full stroke length, the system seems to approach a periodic motion and asymptote to a limiting average speed (see figure 2.2) in the entire parameter range we explored. Although the system is easy to define in terms of the states of the point masses, it is easier to analyze in terms of the Center of Mass (CoM) position $c := (mx + My)/(m + M)$ and the position difference $d := y - x$. In these coordinates it is evident that the actuator does no work on c – it is an "internal force" – and

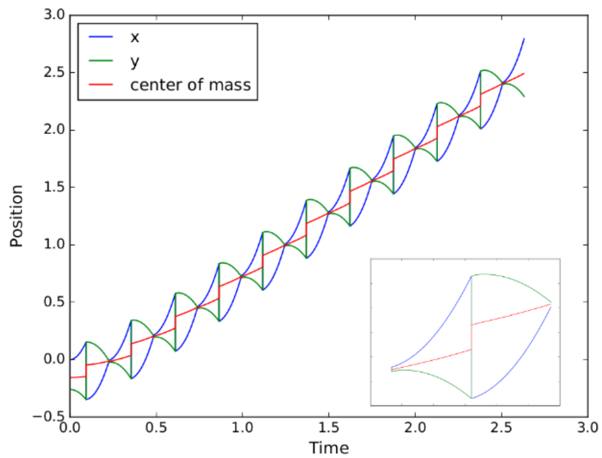

Figure 2.2: Position of m, M, and the center of mass vs. time in an example simulation. Parameter choices followed §2.3. Force $F_t = 45\text{sgn}(x - y)\ N$; dynamic friction coefficient was $\mu = 1$. Figure shows positions of m (dashed line), M (solid line), and center of mass $(mx + My)/(m + M)$ (dotted line) over time. The graph at left-upper corner shows an enlarged cycle starting at $y - x = L_{min}$. The accelerations of m and M are always opposite to each other. They accelerate until the length constraint at L_{min} and L_{max} stops them and they instantly swap their positions. The position swap causes a jump of the center of mass. Note that for the first two cycles of motion the CoM is noticeably changing average speed, which quickly asymptotes to its limiting value.

all work on the CoM is done by the friction forces. Since we are using the Coulomb friction model, once the actuator force is sufficient to get the smaller mass m out of static friction, the actuator will successfully expand and contract to its full extent at any CoM speed. If the motion approaches a limit cycle in $[d, \dot{d}]^\top$, the average speed of the CoM over a long simulation will converge to the speed at the limit cycle, v_{ss} of (2.5).

$$v_{ss} := \frac{\text{Distance traveled in one steady cycle}}{\text{Duration of one steady cycle}} \tag{2.5}$$

For our numerical simulations, we ran the simulations until relative difference of averaged v_{ss} between two consecutive cycles decreased to within 10^{-4}.

2.3 Numerical Simulation Results

We chose simulation parameters appropriate for a small robot or mid-size vertebrate such as a rabbit. We took the masses to be $m = 1$ kg, $M = 1.5$ kg. The mass ratio M/m specifies the instantaneous center of mass displacement when masses are swapped, which becomes $0.2d$ for these values. The minimum and maximum distance between two masses were taken to be $L_{min} = 0.02$ m, and $L_{max} = 0.5$ m respectively. The range of dynamic friction coefficient μ we explored was 0.05 to 1.

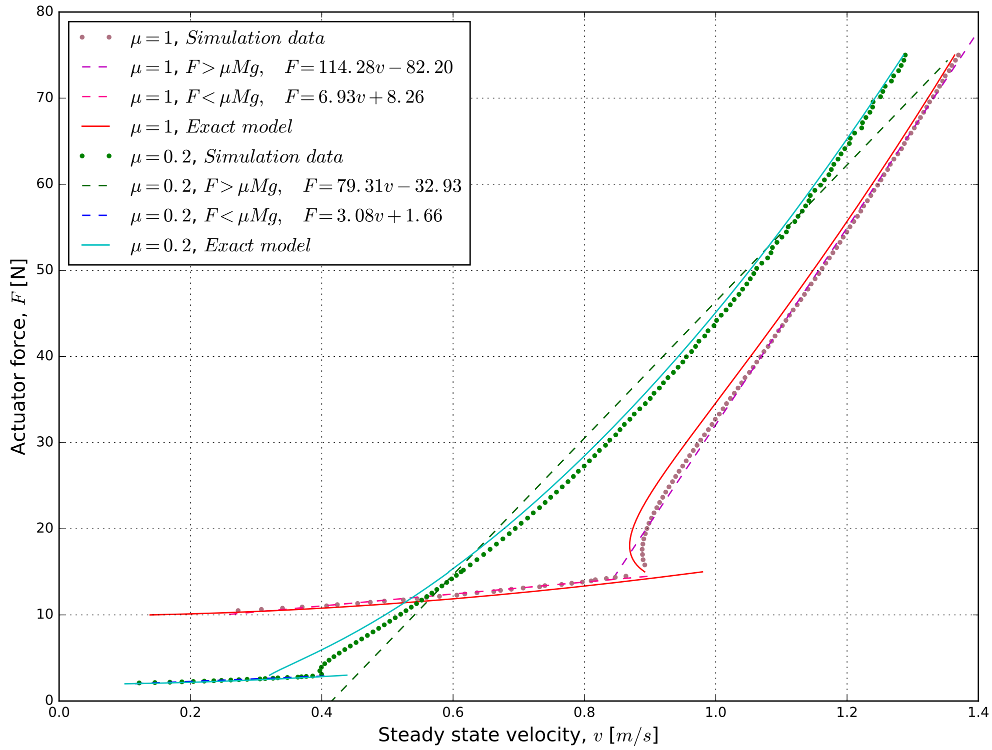

Figure 2.3: Constant force v.s. steady state velocity under Coulomb friction model, $\mu = 1$ and $\mu = 0.2$. We plotted simulation results for Regime I where $F < \mu Mg$, with $\mu = 1$ (black dots) and with $\mu = 0.2$ (grey dots). These are close to their regression lines $\mu = 1$ (dashed black line) $R^2 = 0.99$ and $\mu = 0.2$ (dashed grey line) $R^2 = 0.99$. For Regime II where $F > \mu Mg$, we plotted simulations for $\mu = 1$ (black dots) and $\mu = 0.2$ (grey dots). Each of these follows a nearly linear relationship different from that of the other force regime ($\mu = 1$ dashed black regression $R^2 = 1.0$, $\mu = 0.2$ dashed grey regression $R^2 = 0.99$). We also plotted the exact solution as calculated in §2.4 when $\mu = 1$ (black solid line), $\mu = 0.2$ (grey solid line) for the entire motion regime. Our notable result is that in each regime, the F to v_{ss} relationship can be described fairly accurately as linear, as demonstrated by the high values of regression R^2. Note that intermediate values of μ interpolate those plotted, and were omitted for clarity.

The range of input force F was set from μmg – the minimal force needed to escape static friction – up to $5Mg$ – enough to produce a greater than $2g$ acceleration on the CoM.

In our simulation, we fix the actuator force to be a constant force with periodically changing directions, i.e. $F_t = F \operatorname{sgn}(x - y)$, where F is a constant input force, and x, y are positions of m and M respectively[1]. Without loss of generality, we start at $d = L_{min} > 0$ and thus m is to the right of M, and the masses are pushed apart. Because the masses swap their positions, the actuator is always either pushing or pulling the smaller mass m to right (positive direction), and pushing or

[1] Attempting to allow the force to depend on d always lead to the highest v_{ss} with the force identically maximal, so this was added as an assumption.

pulling the larger mass M to the left (negative direction). Mass M has better traction than m by having larger friction forces, due to its larger normal force on the substrate. Thus, the net friction force generated in this way can move the system in the positive direction.

2.3.1 Two motion regimes arise

We fixed all of the parameters and only change the absolute value of actuator force, F, examining the changes induced in v_{ss}. Two fundamentally different motion regimes arose. Regime I when actuator force is only large enough to take small mass m out of its static friction cone, i.e. $f_m < F \leq f_M$, leading to the small mass moving against a static support offered by the large mass. Regime II when actuator force is large enough to move both small and large masses $F > f_M$, leading to the large mass sliding in the opposite direction of the small mass.

For each input actuator force F, we observed a corresponding final steady state velocity v_{ss} of the system. The figure 2.3 shows the relationship for $\mu = 1$ and $\mu = 0.2$, although a similar qualitative relationship exists for the entire range we explored $0.05 \leq \mu \leq 1$.

For Regime II we explored forces in the range $[\mu Mg, 5Mg]$, which represent a reasonable range of mass specific actuator force density for animals[38, 39, 40], electrically powered autonomous robots, and internal combustion engines[41, 42].

Writing $v_{ss} = kF + b$, we also explored the relationship of k and b with μ following parameter choices in §2.3. The quadratic regression equation for k vs. μ is: $k = -1.9 \cdot 10^{-3}\mu^2 + 11 \cdot 10^{-3}$, with $R^2 = 0.99$. The linear regression equation for b vs. μ is $b = 0.24\mu + 0.46$, with $R^2 = 0.99$.

2.3.1.1 Governing relationship

Combining our numerical results, we found that the averaged steady state velocity versus actuator force model for the system Regime II can be empirically described by a single governing equation correct for actuator forces less than $5Mg$, and $0.05 \leq \mu \leq 1$.

Since the model we used has physical units, to obtain a more principled governing relationship we first switched to non-dimensional form. Let $F := \tilde{f}\mu Mg$ and $v_{ss} := \tilde{v}\sqrt{Lg}$, where $\tilde{f}$ is the non-dimensional force, in units of friction for the large mass, and $\tilde{v}$ is the non-dimensional velocity, in units of the velocity reached in free-fall over the maximal length of the actuator.

The non-dimensional model is shown in (2.6), to two digits precision, and was constructed for $\tilde{f}$ between 1 and $5/\mu$.

$$\tilde{v} = 0.073\mu\left(1 - 0.178\mu^2\right)\tilde{f} + 0.11\mu + 0.20 \tag{2.6}$$

2.4 Exact solution

Assuming the system will achieve a steady state averaged velocity v_{ss}, we solved a closed form solution for v_{ss} versus μ and F. We start from the dynamic motion equations (2.1), (2.2), and recall $d := x - y$. The equations become:

$$\begin{aligned} m\ddot{x} &= F\,\mathrm{sgn}(d) - \mu\, mg\,\mathrm{sgn}(\dot{x}) \\ M\ddot{y} &= -F\,\mathrm{sgn}(d) - \mu\, Mg\,\mathrm{sgn}(\dot{y}) \end{aligned}$$

Suppose, as we empirically observed, that each cycle starts from $d = L_{min}$, and account for the fact that this state appears immediately after a plastic collision, making the velocities of m and M equal. We obtain $\dot{x}_0 = \dot{y}_0 = \dot{c}_0$, where $\dot{c}_0$ is the velocity of CoM. We define, w.l.o.g., that this velocity's direction is positive. Due to the negative force on the large mass M, it first decelerates until its velocity goes to zero, and then accelerates towards negative the direction, or remains zero if F is not large enough to push it out of static friction. The motion is shown schematically in figure 2.4.

$$m\ddot{x} = F - \mu mg \tag{2.7}$$

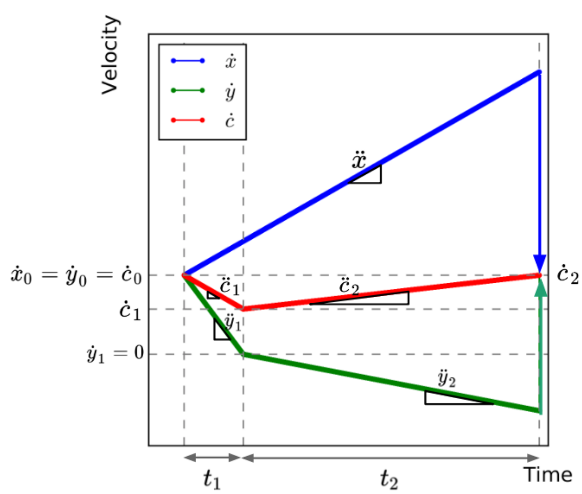

Figure 2.4: Schematic of velocities within a cycle of Regime II motion. We plotted the velocity of m (dashed line), M (solid line) and CoM (dotted line) in one velocity cycle. The initial value of these velocities are $\dot{x}_0 = \dot{y}_0 = \dot{c}_0$. t_1 is the time it takes for the velocity of M starting from $\dot{y}_0$ to decrease to 0. The velocity of the CoM at t_1 is $\dot{c}_1$. t_2 is the remaining time needed for distance between m and M to achieve maximum length bound. $\ddot{x}$ is the constant acceleration of m, and $\ddot{y}_1$, $\ddot{c}_1$, $\ddot{y}_2$, $\ddot{c}_2$ are acceleration of M and CoM during time period t_1, t_2. At the end of the cycle, velocities are reset by a second plastic collision (represented by two arrows).

$$M\ddot{y} = \begin{cases} -F - \mu M g & \dot{y} > 0 \\ \min(-F + \mu M g, 0) & \dot{y} \leq 0 \end{cases} \tag{2.8}$$

The equation of motion for the CoM of the system in the first stage of figure 2.4 is $\ddot{c}_1 = -\mu g$. The time spent at first stage can be calculated from the time needed for M to achieve zero velocity, $t_1 = \mathcal{T}_1(\dot{c}_0) = \dot{c}_0/(F/M + \mu g)$. When $\dot{y} = 0$, the center of mass velocity becomes: $\dot{c}_1 = \dot{c}_0 F/(F + \mu M g)$.

Time t_2 is the time needed for the distance between m and M to reach the maximum length bound. It can be determined by a second order equation (2.9), and the result of $t_2 = \mathcal{T}_2(\dot{c}_0)$ is a function of initial CoM velocity.

$$L_{max} - L_{min} - \frac{1}{2}(\ddot{x} - \ddot{y}_1)t_1^2 = \dot{x}_{t_1} t_2 + (\ddot{x} - \ddot{y}_2)t_2^2 \tag{2.9}$$

Here $\ddot{y}_1$ is the acceleration of M when $\dot{y} > 0$, and $\ddot{y}_2$ is the acceleration of M when $\dot{y} < 0$ as shown in figure 2.4 (the slope of solid line). They are calculated from (2.8).

Thus, the update rule for CoM velocity after each velocity cycle is:

$$\dot{c}_2 = \frac{F}{F + \mu M g}\dot{c}_0 + \frac{M - m}{M + m}\mu g + \mathcal{T}_2(\dot{c}_0) \tag{2.10}$$

By solving for the fixed point of this equation, we can get $\dot{c}_0$ as a function of F and μ, $\dot{C}_0(F, \mu)$. The averaged steady state CoM velocity v_{ss} can be calculated by (2.11). The second term is the CoM position shift due to instantaneous swaps of two masses at L_{min} and L_{max}.

$$\mathcal{V}_{ss}(F, \mu) = \frac{2F + \mu M g}{2(F + \mu M g)}\dot{C}_0 + \frac{M - m}{m + M}\frac{L_{max} - L_{min}}{2(\mathcal{T}_1 + \mathcal{T}_2)} \tag{2.11}$$

The symbolic calculations were done in Mathematica.

2.4.1 Comparison with governing relationship

We compared our empirically derived governing relationship of §2.3.1.1 with the analytical solution of §2.4. The relative error of the model is below 5% except for small forces and low friction, where it grows to 11%.

Our results are best summarized by the examples in figure 2.3, showing v_{ss} vs. F regressions, numerical model simulation results, and exact algebraic results for both motion regimes. One puzzling observation regarding this figure is the separation between exact and numerical curves. Decreasing integrator time-steps, and requiring more stringent convergence from the numerical simulations does bring these curves closer together — an observation that suggests that convergence to the limit cycle may be quite slow. However, these differences still remain somewhat puzzling.

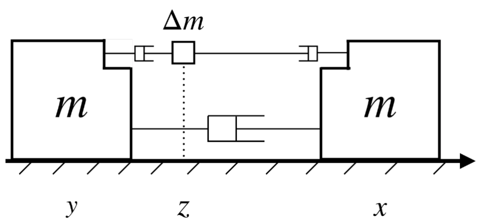

Figure 2.5: Schematic representation of the mechanical model with a swapping mechanism. x, y, z represent the position of each point mass.

2.5 Realistic swapping yields similar results

To make the model more realistic in mimicking an inchworm or a legged robot/animal shuffling, we modeled a mechanistic means for transferring the mass difference $\Delta m := M - m$ between the two point masses of the previous model. To do so we postulated an internal force pushing Δm from one side to the other side, while the distance between the two end-point masses is held constant. This is illustrated in figure 2.5. When the Δm mass reaches its destination it collides in a plastic collision, and all three masses have zero relative velocity post-collision.

The equation of motion remains the same as (2.1) and (2.2) when the system is slipping and Δm is fixed on one of m. As our notation convention, we take the mass m with Δm as M and write its position as y. We denote the other m mass by x, and when Δm is in motion, it is always moving from x to y. From here on we will refer to the two end-point masses, while they are held at constant distance, as the $m - m$ system. The relative movement between Δm and $m - m$ system can be described as follows. Taking $m - m$ as one object, its center is $c := (x + y)/2$. Supposing a force of $F_{\Delta m}$ is used to push Δm (whose position is denoted by z) against $m - m$, the equations of motion for the exchange of Δm are:

$$\begin{aligned} 2m\ddot{c} &= \begin{cases} 0 & \text{if } \dot{c} = 0 \text{ and } |F_{\Delta m}| < \mu g(2m + \Delta m) \\ -F_{\Delta m} - \operatorname{sgn}(\dot{c})\mu g(2m + \Delta m) & \text{else} \end{cases} \\ \Delta m\ddot{z} &= F_{\Delta m} \end{aligned}$$

2.5.1 Numerical simulation

Following the same choice of parameters in §2.3, we use a non-dimensional parameter α to give $F_{\Delta m} := \alpha F \operatorname{sgn}(z - x)$ as the force pushing Δm from M to m, where F the actuation force between m and M. In figure 2.6, we compare the cycle behaviour between instant swap model and constant swapping force models with $\alpha = 1, 5, 10$, in both one mass moving and two mass moving regimes. The models are still asymptotic to a limiting periodic cycle, which vary slightly with α.

2.5.2 Four motion regimes arise

With the parameters selected in §2.3, we found four motion regimes arising from two independent conditions. The first condition was discussed in §2.3.1, namely comparing actuation force F with μmg and μMg. It leads to either one or two masses in dynamic friction during the times $x - y$ changes. The second condition was comparing the total friction force of the system $\mu(2m + \Delta m)g$ with swapping force $F_{\Delta m}$. When moving sufficiently slowly, if $|F_{\Delta m}| \leq \mu g(2m + \Delta m)$, the $m - m$ system first slowed down to a stand-still and stayed in that position until the end of Δm motion. If $|F_{\Delta m}| > \mu g(2m + \Delta m)$, the reaction force for moving Δm was large enough to pull the $m - m$ system out of friction cone during swapping and pushed it backwards. For example, in figure 2.6, when $\alpha = 1$, $m - m$ slowed down and stopped during Δm swapping. When $\alpha = 5, 10$, $m - m$ slowed down and was eventually accelerating towards the negative direction during Δm swapping.

In figure 2.7 we plotted the relationships for $\mu = 0.1, 0.6, 1.0$, and swapping force coefficients $\alpha = 1, 5, 10$. The figure shows that a new "swap with stop" motion regime appears. Outside this new motion regime, the F to v relationship follows the same qualitatively nearly linear relationship that it did for the instantaneous swap model.

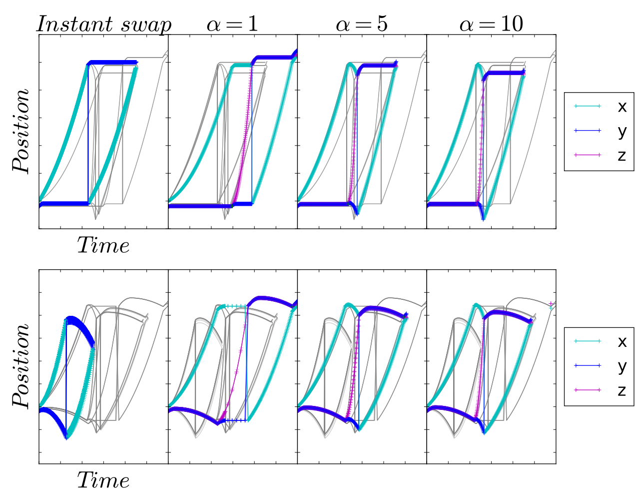

Figure 2.6: Position of m, M and Δm vs. time within a cycle. We plotted these for the one mass moving regime (top, $F = 12N$) and two masses moving regime (bottom $F = 20N$). The dynamic friction coefficient was $\mu = 1$. Figure shows the position of m (dashed line, x), M (solid line, y) and Δm (dotted line, z) plotted with time and position starting at the L_{min} collision. We plotted four cycles, illustrating the rapid convergence to a limit cycle. To facilitate comparison, we plotted all conditions in each subplot with one condition highlighted (darker, with markers) and the others rendered de-emphasized (grey). We compared the instantaneous swap model (left column) and values of $\alpha = 1, 5, 10$ (second to fourth columns). Note that $m - m$ keeps its position during Δm swapping when $\alpha = 1$, whereas it moves backwards when $\alpha = 5, 10$.

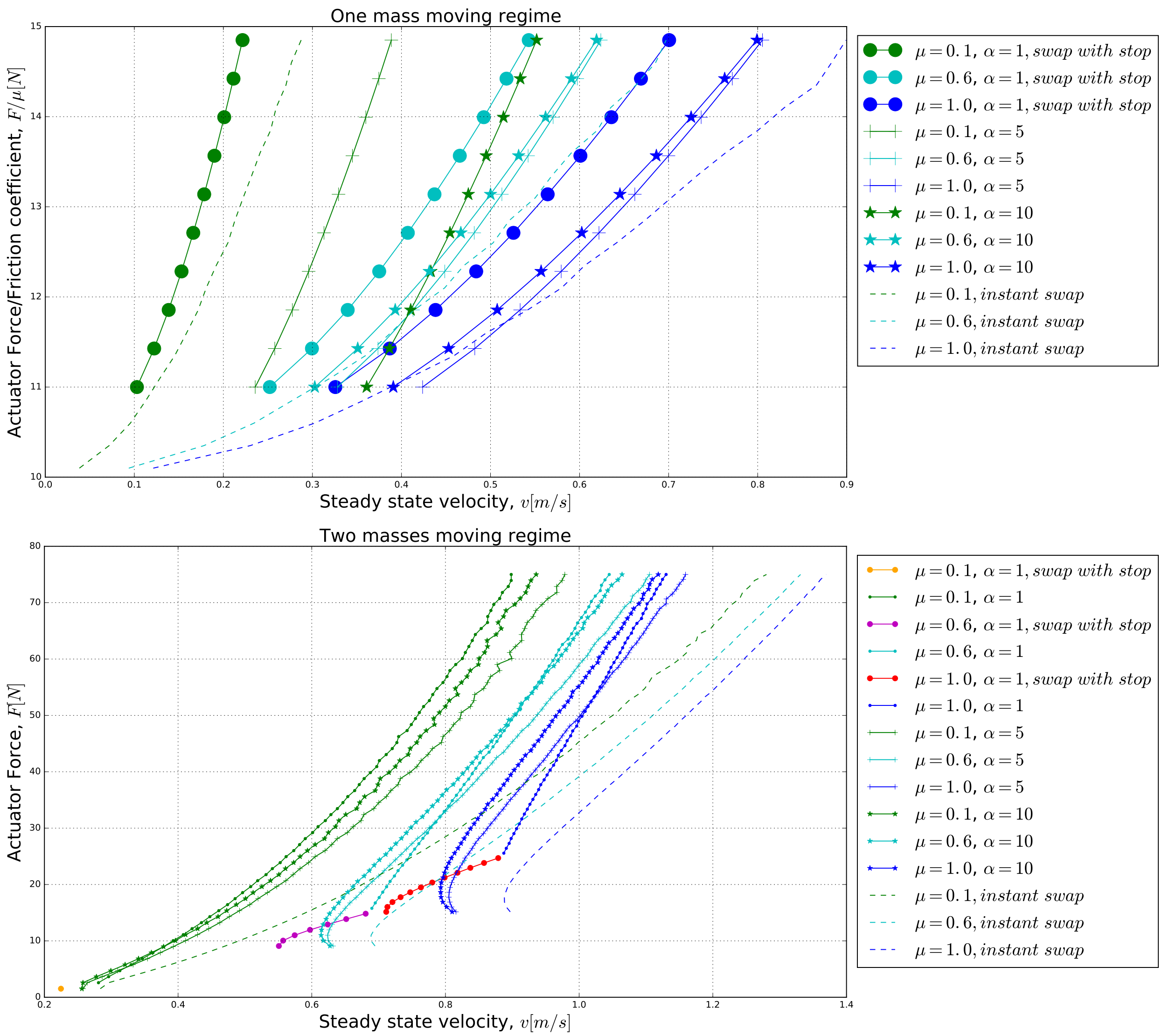

Figure 2.7: Actuation force vs. steady state velocity under Coulomb friction model with constant swapping forces. We plotted the simulation results for friction coefficient $\mu = 1$ (black), $\mu = 0.6$ (thicker grey), and $\mu = 1.0$ (grey). For each μ, we plotted swapping forces $\alpha = 1$ ($\cdot$ marker), $\alpha = 5$ (+ marker), $\alpha = 10$ ($\star$ marker), and instant swap (dashed lines). We also plotted the new "stop during swap" motion regimes (larger round markers with dotted lines). All models are linear to an $R^2 > 0.99$ (single mass moving regime) and $R^2 > 0.96$ (two masses moving regime).

2.6 Discussion

Our simulations and our analytic studies of the models we described brought us to a surprising conclusion: we conclude that the averaged velocity versus input force in Coulomb friction dominated motion is a linear relationship, at least for the parameter range governing small robots and animals. We observed this qualitative relationship to hold both in a simplified "instant-swap" model (§2.2) and a more physical weight redistribution model (§2.5).

The very existence of a functional relationship between force and velocity in a Coulomb friction governed system should come as a surprise, because once static friction is compensated for the Coulomb friction model predicts a speed-independent dynamic friction. Thus, for example, in an idealized wheeled system without air resistance or lubricant viscosity modeled, no upper limit on speed exists. Once the drive motor is strong enough to overcome internal friction, the wheeled system has a constant acceleration rather than a limiting velocity.

The appearance of a viscosity-like relationship out of simple dry friction interactions is somewhat reminiscent of the appearance of viscosity in statistical mechanics, where thermodynamic equilibrium and the associated viscous dissipation appear after only a small number of particle collisions.

It may also be that our observation can be understood in light of a fundamental limit of friction-based locomotion: the CoM can never move faster relative to the ground than the body can change shape. If it did, all points of contact would be slipping in the direction of motion and therefore producing braking forces. Broadly speaking, the speed of shape change is associated with a kinetic energy which must be developed by an actuation force acting over the length of the body. The relationship $mv^2 \propto F \cdot l$ suggests that v should scale as $F^{0.5}$ — which is not whoely incompatible with the lower limit of friction $\mu = 0.1$ that we explored. However, if instead we assumed that the duration of force action is bounded, then $\Delta(mv) \propto F\Delta t$, and the shape change speed is linear with F. As both these limitations play a role in practice, a close to linear relationship may be inevitable regardless of the specifics of the friction model.

CHAPTER 3

Multi-legged Locomotion Model

3.1 Introduction

Multi-legged locomotion has achieved great success with its mobility and stability. Multi-legged robots have the potential to navigate diverse and challenging terrains that are difficult for wheeled or tracked robots. By emulating the legged locomotion of animals, these robots can traverse uneven surfaces, climb stairs, overcome obstacles, and explore environments that are inaccessible to other types of robots. They can achieve increased stability and redundancy in their locomotion, using multiple contacts at the same time. Comparing to bipedal robots, for example Atlas (28 DoF, 6DoF per leg), and quadrupedal robots, MIT cheetah and Boston Dynamics Spot (12 DoF, 3 DoF per leg), multi-legged robots could achieve similar maneuverability and stability using even less actuators. For example, the family of RHex robots [6], RoACH robots [9], Sprawl robots [10], and BigAnt[8] equip with 1DoF or even less per leg. Different from bipedal locomotion, where slipping could induce falling over, and require postural control strategies for slip avoidance [43], some multi-legged locomotion, however, exhibit slipping in a significant portion of motion [44].

How to model multi-legged systems efficiently is still a challenging question due to the complexity of interactions between the body, legs and the substrate. The possible interactions among the groups of legs with the ground could grow combinatorially. When considering the possibility of slipping in multi-legged locomotion, the complexity of modeling increases further. Previous work used template, the simplest model with the least number of variables and parameters, to trim down the complexity and analyze body motion [45]. For example, spring-loaded inverted pendulum (SLIP) and lateral leg spring (LLS) are widely used a template to model the sagittal and horizontal motion of legged systems. However, they do not deal with the interaction among multiple legs nor slipping.

As the number of legs increases and the relative size of each leg becomes smaller, the locomotion of a multi-legged robot converges to that of a slithering of snake robot. Slithering can be understood using geometric mechanics, and is “principally kinematic” in the sense of admitting equations of motion that do not require momentum or velocity states. The equations of motion can be expressed as a "local connection": a linear relationship between the rate of shape change $\dot{r}$ and the body velocity v_b: $v_b = A(r)\dot{r}$ (see e.g. [26, 27]). It is natural to ask how this transition to momentum-free equations of motion occurs with increasing numbers of legs, and whether we can obtain a model for multi-legged locomotion without momentum.

Previous observation showed that with three points of contact, for example the motion of ants and hexapod BigAnt, momentum can be removed and the motion can be described through connection [30]. Previous work presented a data-driven method to efficiently construct this local connection model for systems with no momentum [28]. An extension of the no-momentum case to the quickly decaying momentum case which appears in high friction sliding contact was presented in [29]. Such local connection models are effective for multi-legged animals and robots with and without slipping, at the predicted level of three or more contacts [30]. Furthermore, by replacing the Coulomb friction term which is linear in normal force but non-smooth and non-linear in contact velocity, with a friction ansatz that is bilinear in normal force and contact velocity [8, Chapter 4], one could easily construct the local connection.

Based on the prior work in extending geometric mechanics to multi-legged locomotion, the body motion of multi-legged robots – typically hexapods and octopods – is much easier to model than is often assumed, because despite the complexity of additional contacts and a combinatorial number of contact states, the ensuing simplification that comes from momentum becoming ignorable more than compensates. Even if each leg ends in a multi-toed foot, leading to a total of a few dozen possible contact points, the simulation models remain eminently tractable. In this chapter, we **(1)** present a fast algorithm for modeling multi-legged systems, that accurately estimates the body velocities given the body shape movements, and provides accurate foot contact force estimates. We validate **(2)** this model against several multi-legged robot datasets, including one with force-torque sensors measuring ground contact forces of each foot. We **(3)** demonstrate that it scales favorably with the number of legs and admits easy parallelization in comparison with a state-of-the-art mechanical simulator.

3.1.1 Multi-legged contact modeling

Legged systems generate propulsion forces through contacting with the substrate. Understanding ground contact forces could help with the studying the mobility, maneuverability and stability. By analyzing these forces, researchers can develop control algorithms that help robots maintain stability while navigating uneven terrain or optimizing their performance. Contact estimation through proprioceptive or external sensing is commonly used to model the kinematic and dynamics of biepdal [46] and quadrupedal locomotion [47, 48].

The contact forces are usually unobservable barely from motion tracking in multi-legged systems, and cannot be measured using the commonly used approach of placing a single force plate for the robot to move over as in [49]. Multi-legged system contacts with more than one point, and every closed kinematic loop between body and ground can support an internal force that produces no net body acceleration. A force plate, measuring only the total wrench applied to it, cannot reveal the interplay of all individual foot contact forces, which may trade off in various ways from step to step. The authors of [50] measured the individual ground contact forces of RHex by installing each leg with a 3d force sensor, and developed a control strategy to modulate the difference of normal forces between fore an hind legs. In-situ Calibration of the force-torque sensors on a hexapod turned out to be a non-trivial task, and the detailed approach we used in calibration was explained in Chapter 4.

Rigid body contacts are usually modeled through linear or non-linear complementarity problems with Coulomb friction. However, these problems are usually computationally expensive to solve. To imporve the computational efficiency, MuJoCo [23] recently played an important role in modeling and optimizing robotics behaviors. It used soft contact model to relax the contact constraints, and smooth the interactions, and therefore improve the computation speed.

In the perspective of fluid dynamics, drag is the analogous "contact" forces between body and flow. At low Reynolds number (when viscous forces are large compared to inertial forces), the net motion can be computed through a local connection over the shape space [51]. Resistive force theory successfully modeled movement of animals and robots on dry granular media [52, 53], and the net motion of a system could be computed by local connection [54].

In this chapter, we tried to bridge multi-legged locomotion governed under Coulomb friction with geometric mechanics through a special viscous-Coulombfriction assumption.

3.1.2 Motivating example: Coulomb friction and viscous-Coulombfriction

Coulomb friction contacts do not yield a linear relationship between shape-change velocity and body velocity except for the cases where at least three contacts are in static friction and in general configuration. Consider the 1-dimensional (1D) case (figure 3.1), where a robot constrained to move along the x axis without changing pitch angle, is driven by M legs such that each foot is commanded to move at a potentially different velocity relative to the body v_i^B for $i = 1, \cdots, M$. Further, assume for simplicity that all the feet bear the same normal load N. Denote the friction coefficient as μ, foot velocity in the world frame as v_i^W, and the unknown robot body velocity as v_B.

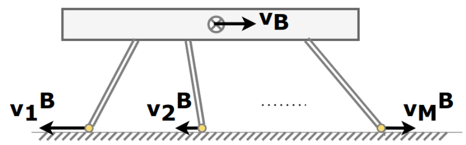

Figure 3.1: A 1-dimensional multi-legged example. Consider two or more equal-height legs, with identical friction properties moving at different horizontal velocities relative to a body constrained to move only in the horizontal direction. The resulting body velocity under Coulomb friction with an odd number of legs is the median leg speed; with an even number of legs the answer is non-unique and any speed between the two median leg speeds will work.

3.1.2.1 Coulomb friction solution is the median

The Coulomb friction acting at each foot is $f_i^C = -\mu N \text{sign}(v_i^W) = -\mu N \text{sign}(v_B + v_i^B)$. At force balance, we want to find v_B, such that

$$0 = \sum_{i=1}^{M} -\mu N \text{sign}(v_B + v_i^B)$$

Without loss of generality, assume the feet are numbered in a monotone order of foot speed. In this 1D example force balance is found when half of the feet are slipping in one direction and the other half are slipping in the opposite direction. If M is odd, assume foot i is moving at the median speed relative to the body. If $v_B = -v_i^B$, that foot will be in static friction, and the remaining feet will be in force balance. Other solutions are possible, depending on the size of the gap between static and dynamic friction, but the median speed foot gives a solution for any plausible choice of friction coefficients. If M is even, assume that the median speed is between v_i^B and v_{i+1}^B. Choosing v_B anywhere in the (open) interval between these velocities will produce force balance. Here additional solutions are also possible if there is a sufficiently large gap between static and dynamic friction coefficients, but the median speeds solution will always exist.

3.1.2.2 Viscous-Coulomb friction solution is linear

Our viscous-Coulomb friction ansatz modifies the contact force equations to be linear in the contact velocity vector. In the 1D case, this becomes $f_i^v = -\mu N v_i^W$. At force balance,

$$0 = \sum_{i=1}^{M} -\mu N (v_B + v_i^B)$$

The immediate solution is $v_B = -\text{mean}(v_i^B)$ – a unique linear solution which in many cases is quite close to the median. The linearity of the viscous-Coulomb friction force-balance solution extends to 2D and full 3D, yielding the local connection.

3.1.2.3 Static friction in 1D is ill-posed

Suppose the system moves without any slipping by coordinating the motions of the legs: $v_1^B = v_2^B = \cdots = v_M^B$. The viscous-Coulomb friction model would correctly and robustly estimate the body velocity $v_B = -v_1^B$, whereas the Coulomb friction model would be at a singularity with an ill-posed numerical problem for $|v_i^W| \to 0$. Any level of noise in the velocities would change the results.

3.1.2.4 Extending to 2D

In 2D, the Coulomb friction becomes $f_i^C = \mu N v_i^W / ||v_i^W||$; our viscous-Coulomb friction ansatz is again $f_i^v = -\mu N v_i^W$. In figure 3.2, we show the total force of a four legged robot slipping under Coulomb or viscous-Coulomb friction. The two friction models produce very similar result when solving for the body velocity under force-torque balance. However, the Coulomb friction model produces a force contour with discontinuities, whereas, the viscous-Coulomb model produces a simple quadratic force contour.

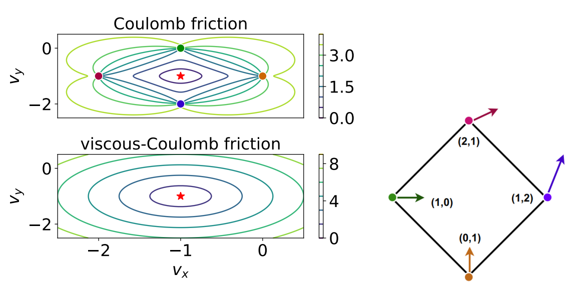

Figure 3.2: Visualization of a 2D case. Consider a 4-legged robot (right plot) whose feet move relative to the body with known velocities (arrows, right plot). We plotted contours of the magnitude of the total force on the body under the assumption that the body is moving at body velocity (V_x, V_y) without rotating, both under Coulomb friction (top left) and our viscous-Coulomb ansatz (bottom left). For Coulomb friction, this function has point discontinuities at those velocities that put any of the feet into static friction (colored dots, color matched across sub-figures). For each friction model, we also indicated the body velocity at force-moment balance (red stars). The plots demonstrate that the viscous-Coulomb ansatz gives rise to a simple quadratic, whereas Coulomb friction produces an almost everywhere smooth function with point discontinuities, yet both produce very similar solutions.

3.2 Model Overview

We propose an algorithm to estimate world frame body velocity from its body shape and shape-changing velocity at the current time frame. Our algorithm takes as inputs: (1) the positions q_j and velocities $\dot{q}_j$ of the robot's feet in the body frame; (2) the spring stiffness k_j of each leg; (3) the friction coefficients μ_j and friction anisotropy $w_{xy,j}$. As outputs it provides: (1) body height $p_{z,0}$, pitch β, and roll α slopes; (2) body velocities $\dot{p}_{xy,0}$ and $\dot{\theta}$; (3) 3D forces at the feet F_j.

The model assumes the robot's (1) pitch and roll angles are small; (2) pitch, roll, and height vary slowly, so their derivatives can be assumed to be zero; (3) the system's inertia is irrelevant since the system is always at force-moment balance; (4) the legs are short and the ground is horizontal at the point of contact of each foot. A summary of the algorithm is shown in algorithm 1. The Python code implementation is open source together with experimental dataset in [31].

Algorithm 1: Pseudo-Code of §3.3

INPUTS: 3D foot positions $\{q_j\}$, velocities $\{\dot{q}_{xy,j}\}$
PARAMETERS (CONSTANTS): mass M, stiffnesses $\{k_j\}$, friction coefficients $\{\mu_j\}$, anisotropies $\{w_{xy,j}\}$

Define function: $\mathcal{N}(s)$ set of legs on ground at state s, $\mathcal{N}(s) := \{k | p_{z,k}(s) < 0\}$, with $p_{z,k}(s)$ from eqn. (3.6)

Solve for force-balance z^* using M, $\{k_j\}$, and $\{q_j\}$ according to §3.3.1; $s_0 \leftarrow (0, 0, z^*)$;
Loop
  if $|\mathcal{N}(s_0)| < 3$, *i.e. 1 or 2 feet on ground* **then**
    Apply §3.3.3: rotate the body as dictated by the gravity using eqn. (3.11), eqn. (3.12), giving s_0^*;
  else
    Solve s_0^* using force and moment balance equations: (3.9);
    if $\mathcal{N}(s_0^*) = \mathcal{N}(s_0)$, *i.e.* s_0^* *has same contacts as* s_0 **then break** from loop;
    Apply §3.3.2: let s_1 be the first point on the ray from s_0 to s_0^* at which contact legs change; $s_0^* \leftarrow s_1$;
  end
  Update current state $s_0 \leftarrow s_0^*$;
end
Compute normal (vertical) forces $F_{z,k}(s_0^*)$ with eqn. (3.7);
Apply §3.4.2 at state s_0^* to solve $\dot{p}_{xy,0}$, $\dot{\theta}$ from eqns. (3.16), (3.18), using $\{q_{xy,k}\}$, $\{\dot{q}_{xy,k}\}$, $\{\mu_k\}$, and $\{w_{xy,k}\}$;
also computed $\{p_{xy,k}\}$, $\{\dot{p}_{xy,k}\}$, and $\{F_{xy,k}\}$ the tangent (horizontal) forces of eqn. (3.15)

RETURN body state $s_0^* = (p_0, \theta, \alpha_x, \alpha_y)$, body velocities $\dot{p}_{xy,0}$, $\dot{\theta}$, and 3D forces at feet $\{F_k\}$;

The algorithm is composed of two steps: (A) find which feet are in contact with the ground and estimate their gravity-induced loading using a spring support model; (B) use force and torque balance to construct the linear local connection model, invert it, and use that to estimate the planar body velocity.

The inputs to the spring support model (A) are: (1) the 3D positions of the feet q_j in the body frame; (2) the spring stiffness k_j of each leg. The outputs of the spring support model are: (1) body height $p_{z,0}$, pitch β, and roll α slopes; (2) gravity-induced loading on each foot $F_{z,j}$ and, implicit in that, which feet are in contact with the ground.

Once the contacting feet are known, we solve for force and moment equilibrium using a viscous-Coulomb friction ansatz which is bi-linear in $F_{z,j}$ and the foot sliding velocities in the world frame $\dot{p}_{xy,j}$, providing an local connection model (B). The inputs to the connection model are: (1)the 2D positions $q_{xy,j}$ and velocities $\dot{q}_{xy,j}$ of the feet in the body frame; (2) the friction coefficients μ_j and friction anisotropy $w_{xy,j}$; (3) the gravity induced loading $F_{z,j}$ computed in (A). The outputs of this local connection model are body velocities: (1) body velocities $p_{xy,0}$ and $\dot{\theta}$; (2) 2D traction forces at the feet $F_{xy,j}$.

Suppose we are given a system with N legs (or other contacts), indexed by $j = 1 \ldots N$. The time-varying foot positions in the body frame of reference are given by $q_j \in \mathbb{R}^3$, $j = 1 \ldots N$. We assume the transformation from body frame to world frame is given by a time-varying rigid body transformation $\Omega \in SE(3)$. The world frame foot positions p_j is

$$p_j := \Omega q_j \tag{3.1}$$

Let p_0 represent the origin of the body frame at the CoM. The rigid body transformation written in homogeneous transformation representation is:

$$\Omega = \begin{bmatrix} \mathsf{R} & p_0 \\ 0 & 1 \end{bmatrix}, \qquad \mathsf{R} \in SO(3)$$

Using Euler angle with roll (α) pitch (β) yaw (θ) representation,

$$\mathsf{R} = \mathsf{R}_z(\theta)\mathsf{R}_y(\beta)\mathsf{R}_x(\alpha) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

We assume a simplified form of R', where pitch and roll angles are small, so they could be approximated by their first order Taylor expansion $\sin(\alpha) \approx \alpha, \cos(\alpha) \approx 1$ and $\sin(\beta) \approx \beta, \cos(\beta) \approx$

1. The first order approximation of R becomes:

$$\mathsf{R}' = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \beta \\ 0 & 1 & 0 \\ -\beta & 0 & \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & \alpha\sin(\theta) + \beta\cos(\theta) \\ \sin(\theta) & \cos(\theta) & -\alpha\cos(\theta) + \beta\sin(\theta) \\ -\beta & \alpha & 1 \end{bmatrix} \tag{3.2}$$

We further assume the rigid body motion is only time-varying in the horizontal plane, i.e. α β and $p_{z,0}$ vary so slowly that their derivatives can be approximated by zeros, $\dot{\alpha} = \dot{\beta} = \dot{p}_{z,0} = 0$. Furthermore, we assume the robot body is low to the ground, so we could ignore the projection from body z-axis to the horizontal plane in world frame. Specifically, Ω' is:

$$\Omega' := \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & p_{x,0} \\ \sin(\theta) & \cos(\theta) & 0 & p_{y,0} \\ -\beta & \alpha & 1 & p_{z,0} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Because of these simplifying assumptions, we can decouple the movements in the xy plane, and the physical units of vertical and horizontal length are decoupled. We use the planar rotation $\mathrm{R}_\theta := \left[\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right]$, and $\mathrm{S} := \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$, together with the fact $\dot{\mathrm{R}}_\theta = \dot{\theta}\mathrm{S}\mathrm{R}_\theta = \dot{\theta}\mathrm{R}_\theta\mathrm{S}$, to represent foot position and velocity in world frame xy plane with:

$$p_{xy,j} = \mathrm{R}_\theta q_{xy,j} + p_{xy,0} \tag{3.4}$$
$$\dot{p}_{xy,j} = \dot{\mathrm{R}}_\theta q_{xy,j} + \mathrm{R}_\theta \dot{q}_{xy,j} + \dot{p}_{xy,0} = \mathrm{R}_\theta\left(\dot{\theta}{\mathrm{R}_\theta}^{-1}\mathrm{S}\mathrm{R}_\theta q_{xy,j} + \dot{q}_{xy,j}\right) + \dot{p}_{xy,0} \tag{3.5}$$

The decoupled foot position in world frame vertical $z-$axis is:

$$p_{z,j} = -\beta q_{x,j} + \alpha q_{y,j} + q_{z,j} + p_{z,0} \tag{3.6}$$

3.3 Spring Support Model : finding the contacts

In this section, we show how to decouple the roll, pitch, and vertical (z-axis) motion of the robot, determine which legs are in contact with the ground, and what supporting force each leg generates. We model the robot as a "body plane", with each leg assumed to be a vertical spring attached to this plane. We assume the system is at force and moment balance about its center, i.e. the origin of

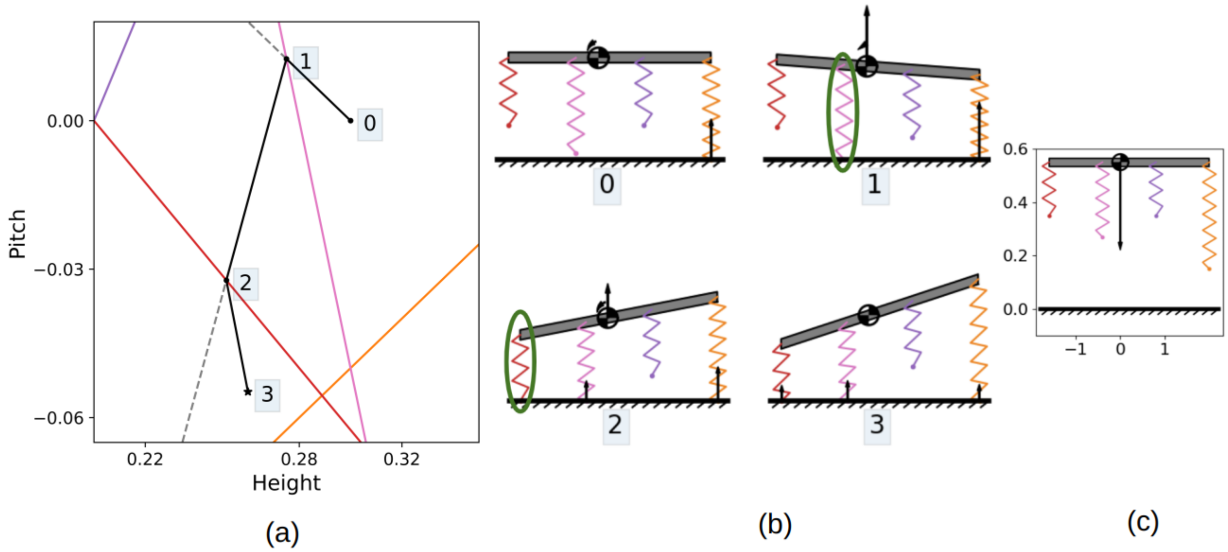

Figure 3.3: Visualization of the search for contact state in a 2D "robot". We indicated the height and pitch ($p_{z,0}$,β) states searched (labels 0-3 of (a)), and visualized the pose and contacts of the "robot" in each of these states (corresponding labels of plots in (b)). Each "robot" leg (zigzag lines in (b)) defines a corresponding co-dimension 1 plane (line here) in (z_0,β) at which it contacts the ground (colored lines in (a) with color same as the leg in (b), (c)). At a $p_{z,0}$ above the plane, the leg is in the air; below it, the leg will be in contact and generate normal forces. With each state being searched (number label in (a)), there is a closed-form solution of the force equilibrium, which we connect to that state with a line interval (black in (a)). If the equilibrium lies in the same contact state the algorithm terminates (star; step 3). Otherwise, the portion of the line segment in another contact state is counter-factual (black dashed in (a)). Instead, we switch to the new contact state and solve it again. Each such transition between contact states lies on a plane corresponding to the leg that made contact (black dot in (a); circled leg in (b)).

the robot frame. A simplified version of this model, without accounting for roll and pitch, can be found in [8]. A similar spring-leg model was used to study legged animals and robots in [55] and later in [14], but they did not specify how to determine which legs are in contact.

We used a 2D "robot" in xz-plane to visually illustrate our search algorithm in figure 3.3. In this 2D case, the algorithm searches for robot height and pitch using foot position in x, z coordinates. The 3D model, derived with details in this section, extends the contact switching lines in 2D searching space to planes in 3D, and its visualization can be found in figure 3.4.

Consider a roll, pitch and height state $s = (\alpha, \beta, p_{z,0})$. From (3.1), we have

$$p_{z,j}(s) := -\beta q_{x,j} + \alpha q_{y,j} + q_{z,j} + p_{z,0}$$

Taking 0 to be the ground level, and up being the positive z-axis direction, legs with $p_{z,j} < 0$ are in contact with the ground. Assuming the normal supporting force $F_{z,j}(s)$ is linearly dependent on $p_{z,j}$, we define the individual leg normal force and the resulting planar moment function by,

$$F_{z,j}(s) := \begin{cases} -K_j \; p_{z,j}(s) & \text{if } p_{z,j}(s) < 0 \\ \\ 0 & \text{otherwise} \end{cases}$$
$$M_{x,j}(s) := -q_{y,j} F_{z,j}(s) \qquad M_{y,j}(s) := q_{x,j} F_{z,j}(s), \tag{3.7}$$

and we denote the total force and moment by,

$$F_z(s) = \sum_{i=1}^{N} F_{z,j}(s), \quad M_x(s) = \sum_{i=1}^{N} M_{x,j}(s),$$
$$M_y(s) = \sum_{i=1}^{N} M_{y,j}(s). \tag{3.8}$$

3.3.1 Finding an initial z^*

Without loss of generality, we number $p_{z,j}$ in non-increasing order. When $\alpha = \beta = 0$, the total normal force at height z such that $p_{z,N_k} \leq -z < p_{z,N_k+1}$ is $F_z([0,0,z]) = \sum_{j=1}^{N_k} K_j(z + p_{z,j})$. We scan values of z, $z := -p_{z,N_k}$,starting with $N_k = 1$, where only the lowest foot is in contact. We increase N_k until $F_z([0,0,-p_{z,N_k}]) \leq Mg < F_z(0,0,-p_{z,N_k+1})$, and then linearly solve using the slope K_{N_k+1} to find z^* such that force balance is achieved. For that z^*, legs $k = 1 \ldots N_k$ are in contact with the ground. Throughout this chapter, we use the index k to vary only over legs that in contact with the ground based on this criterion, and by $F_{z,k}$ the normal force of those legs.

3.3.2 Solving for state

Next, we solve for the full state, s, containing the small pitch and roll angles, and the body height, maintaining vertical force balance and moment balance of the moments generated by the normal forces, i.e. $F_z - Mg = M_x = M_y = 0$. We start with the initial condition $s_0 = (0,0,z^*)$, with $F_z = Mg$. Taking α,β, $p_{z,0}$ as unknowns and holding the legs in contact constant, i.e. assuming the change in state s does not generate or break contacts, these values are a solution s_0^* for a 3-dimensional linear system. When there are three or more legs in contact, the linear system to solve for s_0^* at normal force and planar moment balance is:

$$\begin{bmatrix} \sum_{k=1}^{N_k} F_{z,k} \\ \sum_{k=1}^{N_k} M_{xy,k} \end{bmatrix} = \begin{bmatrix} -Mg \\ \mathbf{0} \end{bmatrix} \tag{3.9}$$

$$\sum_{k=1}^{N_k} \begin{bmatrix} -K_k q_{x,k}\beta + K_k q_{y,k}\alpha + K_k p_{z,0} + K_k q_{z,k} \\ K_k q_{x,k} q_{y,k}\beta - K_k q_{y,k}^2\alpha - K_k q_{y,k} p_{z,0} - K_k q_{y,k} q_{z,k} \\ -K_k q_{x,k}^2\beta + K_k q_{x,k} q_{y,k}\alpha + K_k q_{x,k} p_{z,0} + K_k q_{x,k} q_{z,k} \end{bmatrix} = \begin{bmatrix} -Mg \\ 0 \\ 0 \end{bmatrix}$$

$$\sum_{k=1}^{N_k} \begin{bmatrix} -K_k q_{x,k} & K_k q_{y,k} & K_k \\ -K_k q_{x,k} q_{y,k} & K_k q_{y,k}^2 & -K_k q_{y,k} \\ -K_k q_{x,k}^2 & K_k q_{x,k} q_{y,k} & K_k q_{x,k} \end{bmatrix} \begin{bmatrix} \beta \\ \alpha \\ p_{z,0} \end{bmatrix} = \begin{bmatrix} -Mg - \sum_{k=1}^{N_k} K_k q_{z,k} \\ \sum_{k=1}^{N_k} K_k q_{y,k} q_{z,k} \\ -\sum_{k=1}^{N_k} K_k q_{x,k} q_{z,k} \end{bmatrix}$$

We check whether the legs in contact at s_0^*, the solution to the above 3-dimensional linear system, are the same as in the initial condition s_0; if so, then s_0^* is the result of our model. If not, then we search along the line segment starting at s_0 and ending at s_0^* for the first change in contacts, which must therefore occur on a plane describing the contact condition for the first leg which would change contact state going along this line segment. This transition point is taken as s_1, and the process repeats for the new legs in contact. Because contact forces are zero on the corresponding contact condition plane, F_z, M_x, M_y are continuous through the change in contacting legs.

In general, let the initial state be $s_0 = [\alpha_0, \beta_0, z_0]$, and the solution to be $s_0^* = [\alpha_0^*, \beta_0^*, z_0^*]$. When the contact indices are different at the solution s_0^* from the initial condition s_0, we search for the next state s_1 along the line segment connecting those two points, $s_1 = s_0 + t(s_0^* - s_0)$, and stop at the first contact transition surface. The contact transition surface for a leg j is when its $p_{z,j}(s_1) = 0$, i.e. it touches the ground but has zero compression. The line segment intersects with such surface when $t = t_j$, where t_j is the solution to

$$p_{z,j}(s_1) = -((1-t_j)\beta_0 + t_j\beta_0^*)q_{x,j} + ((1-t_j)\alpha_0 + t_j\alpha_0^*)q_{y,j} + q_{z,j} + (1-t_j)z_0 + t_j z_0^* = 0$$

To find the first contact transition, we solve for all intersections t_j along the line segment, and take the one that is the closest to the initial condition:

$$\begin{aligned} t &= \min_{j=1\cdots N, t_j\in[0,1]} \{t_j\} \\ &= \min_{j=1\cdots N, t_j\in[0,1]} \left\{ \frac{\beta_0 q_{x,j} - \alpha_0 q_{y,j} - q_{z,j} - z_0}{-(\beta_0^* - \beta_0)q_{x,j} + (\alpha_0^* - \alpha_0)q_{y,j} + (z_0^* - z_0)} \right\} \end{aligned}$$

Let $\delta s = s_0^* - s_0 = [\delta\alpha, \delta\beta, \delta p_{z,0}]$,

$$t = \min_{j=1\cdots N, t_j \in [0,1]} \left\{ \frac{\beta_0 q_{x,j} - \alpha_0 q_{y,j} - q_{z,j} - z_0}{-\delta\beta q_{x,j} + \delta\alpha q_{y,j} + \delta z} \right\} \tag{3.10}$$

3.3.3 Case of too few contacts

As the search iterates, we may encounter a state where only one or two legs are in contact, and the linear force-torque balance equation becomes under-determined. When there are fewer than three legs in contact, the CoM, located at the origin, generates a moment around the contact point(s) and we tilt the body plane, i.e. change β and α, approximating the rotation this moment would induce until reaching an angle where an additional leg contacts the ground.

We proceed to describe the tilting directions as if they were rotations with an axis. When only one leg p_1 is in contact, the rotation is in the vertical plane containing the leg and the COM, around the contact point, while holding the compressed length of the contact leg to be a constant. Let the change towards next state be $\delta s = [\delta\alpha, \delta\beta, \delta p_{z,0}]$. It needs to satisfy (1) The compressed length remains the same: $p_{z,1}(s) = p_{z,1}(s + \delta s)$; (2) The CoM remains in the vertical plane passing through the contacting point p_1, its hip $(p_{x,1}, p_{y,1}, 0)$, and the origin. Therefore, we get the state changing direction as

$$\delta s = t[p_{y,1}, -p_{x,1}, -p_{y,1}^2 - p_{x,1}^2]. \tag{3.11}$$

The magnitude of state change t is the minimum amount that results in another contact leg, which can be solved the similar to eqn. (3.10), with constraint $t > 0$.

When two legs p_1, p_2 are in contact, the rotation is around the line connecting their contact points, and in the direction of the moment, generated by the gravity around this line. Let $\delta x = p_{x,1} - p_{x,2}, \delta y = p_{y,1} - p_{y,2}, c = p_{y,1}p_{x,2} - p_{y,2}p_{x,1}$, the direction of the change in states is

$$\delta s = -t[\delta y, \delta x, c]c/|c|. \tag{3.12}$$

Again, the magnitude of state change t is the minimum amount that results in another contact leg, which can be solved the similar to eqn. (3.10), with constraint $t > 0$.

3.4 Local connection model : modeling the friction

After knowing which legs are in contact and their gravity loading we solve for the body planar velocity $(\dot{p}_{xy,0}, \dot{\theta})$, obtained by imposing force and moment balance. While classical approaches suggest Coulomb friction is the correct tribological model for sliding dry contacts, we show that a viscous-Coulomb ansatz which is bilinear in both loading force and sliding velocity makes for a

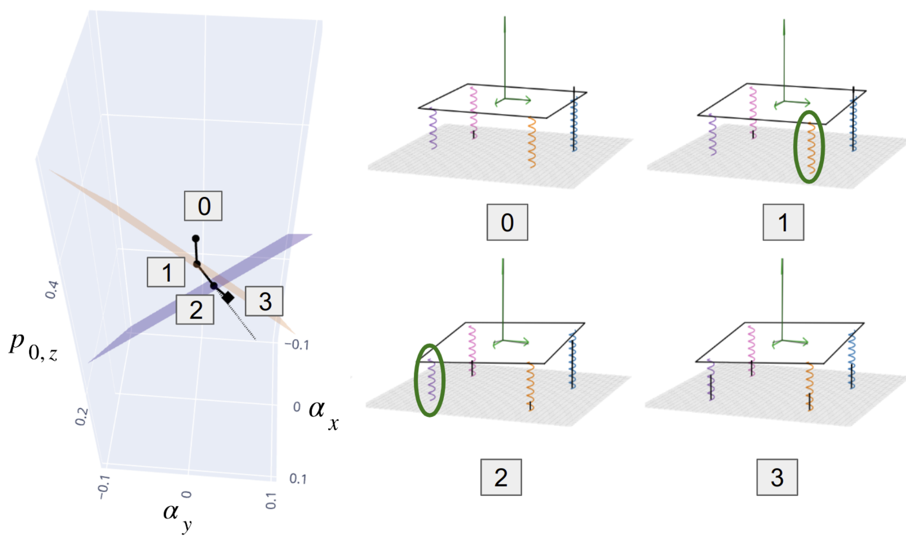

Figure 3.4: Visualization of the search for contact state in a 3D "robot" (labels 0-3, left). We indicated the searching states as roll, pitch and height ($\alpha, \beta, p0$), and visualized the pose and contacts of the "robot" in each iteration (labels 0-3, right). Each "robot" robot leg (spiral lines, right) defines a corresponding co-dimension 2 plane in state space (left), at which it contacts the ground as the same color as plotted on the "robot". The search starts with the initial condition, where z-axis force balance is satisfied. Then it solves force and moment balance for pitch, roll and height, where the grey dashed line is the direction of the solution given current contact indices. It goes along that direction, and stops at the first intersection with a color line, where a leg switches its contact state. The iteration continues, until a solution is found within its current polygon, i.e. contact indices do not change. The green circle highlights the legs switching contact at that iteration.

linear system of equations that leads to a local connection model.

3.4.1 Friction forces

The classical approaches to mechanics suggest that the contact between foot and ground should be modeled by Coulomb friction (middle term below), defining the use of H_k

$$F_{xy,k} = -\frac{\dot{p}_{xy,k}}{\|\dot{p}_{xy,k}\|}\mu_k F_{z,k} = \mathrm{H}_k \dot{p}_{xy,k}. \tag{3.13}$$

The choice of $\mathrm{H}_k = -\mu_k F_{z,k}/\|\dot{p}_{xy,k}\|$ would provide equality, i.e. Coulomb friction, but this would also produce the well-known problem of singularity at $\dot{p}_{xy,k} = 0$. Let $v_k := \|\dot{p}_{xy,k}\|$ be the magnitude of the sliding velocity. We explore the tractability of alternative friction models using

$$\mathrm{H}_k := -\mu_k F_{z,k}\frac{\varepsilon + v_k}{\varepsilon + v_k^2}. \tag{3.14}$$

When $\varepsilon \to 0$, H_k approaches that of the Coulomb friction model; when $\varepsilon \to \infty$, $\mathrm{H}_k \to -\mu_k F_{z,k}$, the friction force becomes $F_{xy,k} = -\mu_k F_{z,k}\dot{p}_{xy,k}$, a combination of viscous and Coulomb friction, which bilinear in slipping rate and normal force.

We further deconstruct (3.13) in terms of body velocity $\dot{\theta}$, $\dot{p}_{x,0}$, and $\dot{p}_{y,0}$ using frame transformations in (3.5):

$$\begin{aligned} F_{xy,k} &= \mathrm{H}_k \left(\mathrm{R}_\theta \left[\dot{\theta}\mathrm{R}_\theta{}^{-1}\mathrm{S}\mathrm{R}_\theta q_{xy,k} + \dot{q}_{xy,k}\right] + \dot{p}_{xy,0}\right) \\ &= \left(\mathrm{H}_k\mathrm{S}\mathrm{R}_\theta q_{xy,k}\right)\dot{\theta} + \mathrm{H}_k\,\dot{p}_{xy,0} + \left(\mathrm{H}_k\mathrm{R}_\theta\dot{q}_{xy,k}\right) \end{aligned} \tag{3.15}$$

3.4.2 Solving for planar body velocity

At quasi-static equilibrium, the system achieves horizontal plane force and moment balance, i.e. $\sum F_{x,k} = \sum F_{y,k} = 0$ and $\sum M_{z,k} = 0$. From planar force balance, using (3.15), we obtain two equations in $\dot{\theta}$, $\dot{p}_{x,0}$, and $\dot{p}_{y,0}$

$$0 = \sum_{k=1}^{N_k} F_{xy,k} = \left(\sum_{k=1}^{N_k} \mathrm{H}_k\mathrm{S}\mathrm{R}_\theta q_{xy,k}\right)\dot{\theta} + \left(\sum_{k=1}^{N_k}\mathrm{H}_k\right)\dot{p}_{xy,0} + \left(\sum_{k=1}^{N_k}\mathrm{H}_k\mathrm{R}_\theta\dot{q}_{xy,k}\right) \tag{3.16}$$

Let $\bar{p}_{xy,k} := p_{xy,k} - p_{xy,0}$, and by (3.4), we have $\bar{p}_{xy,k} = \mathrm{R}_\theta q_{xy,k}$. The z-moment exerted by a leg about the body center is given by:

$$M_{z,k} = \bar{p}_{xy,k}^{\mathsf{T}}\mathrm{S}F_{xy,k} = \left(\bar{p}_{xy,k}^{\mathsf{T}}\mathrm{S}\mathrm{H}_k\mathrm{S}\mathrm{R}_\theta q_{xy,k}\right)\dot{\theta} + \left(\bar{p}_{xy,k}^{\mathsf{T}}\mathrm{S}\mathrm{H}_k\right)\dot{p}_{xy,0} + \left(\bar{p}_{xy,k}^{\mathsf{T}}\mathrm{S}\mathrm{H}_k\mathrm{R}_\theta\dot{q}_{xy,k}\right) \tag{3.17}$$

Giving the third and final equation:

$$0 = \sum_{k=1}^{N_k} M_{z,k} = \left(\sum_{k=1}^{N_k} \bar{p}_{xy,k}^{\mathsf{T}} \mathrm{S}\mathrm{H}_k \mathrm{S}\mathrm{R}_\theta q_{xy,k}\right)\dot{\theta} + \left(\sum_{k=1}^{N_k} \bar{p}_{xy,k}^{\mathsf{T}} \mathrm{S}\mathrm{H}_k\right)\dot{p}_{xy,0} + \left(\sum_{k=1}^{N_k} \bar{p}_{xy,k}^{\mathsf{T}} \mathrm{S}\mathrm{H}_k \mathrm{R}_\theta \dot{q}_{xy,k}\right) \tag{3.18}$$

In the case when $\varepsilon \to \infty$, H_k being rate $\dot{p}_{xy,k}$ independent, the three force and moment balance equations are linear in the body velocity $\dot{p}_{xy,0}$, $\dot{\theta}$ and foot velocity in body frame $\dot{q}_{xy,k}$. One could solve the system by 3d matrix inversion.

3.4.3 Anisotropic friction

In addition to classic Coulomb friction and viscous friction, we consider the possibility that H_k can be dependent on the system's configuration (q), modeling forces generated by a wheel, skate, claw, or otherwise non-isotropic frictional contact. We consider an anisotropic viscous friction model, where H_k is a symmetric positive semidefinite matrix, $\mathrm{H}_k(q) := \mathrm{R}_\theta \mathrm{H}_{q,k}(q) \mathrm{R}_\theta{}^{-1}$ taken to be independent of $\dot{p}_{xy,k}$, but (possibly non-linearly) dependent on all elements of q. We assume that each contact is associated with an enhanced traction direction and associated magnitude, expressed in body coordinates as a vector $w_{xy,k}$, defined as:

$$\mathrm{H}_k := -\mu_k F_{z,k} \mathrm{R}_\theta (\mathrm{I}_2 + w_{xy,k} w_{xy,k}^{\mathsf{T}}) \mathrm{R}_\theta{}^{-1} \tag{3.19}$$

This changes the circular cross-section of the friction cone into an ellipsoidal one. Even with this dependence, the equations (3.18) and (3.16) are still linear in the velocities $\dot{p}_{xy,0}$, $\dot{\theta}$ and $\dot{q}_{xy,k}$. The planar force exerted by a leg becomes

$$F_{xy,k} = \left(\mathrm{R}_\theta \mathrm{H}_{q,k} \mathrm{S} q_{xy,k}\right)\dot{\theta} + \mathrm{R}_\theta \mathrm{H}_{q,k} \mathrm{R}_\theta{}^{-1}\, \dot{p}_{xy,0} + \left(\mathrm{R}_\theta \mathrm{H}_{q,k} \dot{q}_{xy,k}\right)$$

The $z-$moment exerted by a leg:

$$\begin{aligned} M_{z,k} &= \left(\bar{p}_{xy,k}^{\mathsf{T}} \mathrm{S}\mathrm{R}_\theta \mathrm{H}_{q,k} \mathrm{S} q_{xy,k}\right)\dot{\theta} + \left(\bar{p}_{xy,k}^{\mathsf{T}} \mathrm{S}\mathrm{R}_\theta \mathrm{H}_{q,k} \mathrm{R}_\theta{}^{-1}\right)\dot{p}_{xy,0} + \left(\bar{p}_{xy,k}^{\mathsf{T}} \mathrm{S}\mathrm{R}_\theta \mathrm{H}_{q,k} \dot{q}_{xy,k}\right) \\ &= \left(q_{xy,k}^{\mathsf{T}} \mathrm{S}\mathrm{H}_{q,k} \mathrm{S} q_{xy,k}\right)\dot{\theta} + \left(q_{xy,k}^{\mathsf{T}} \mathrm{S}\mathrm{H}_{q,k} \mathrm{R}_\theta{}^{-1}\right)\dot{p}_{xy,0} + \left(q_{xy,k}^{\mathsf{T}} \mathrm{S}\mathrm{H}_{q,k} \dot{q}_{xy,k}\right) \end{aligned}$$

The linear system to solve for planar force and moment balance is:

$$
\begin{aligned}
\begin{bmatrix} \sum_{k=1}^{N_k} F_{xy,k} \\ \sum_{k=1}^{N_k} M_{z,k} \end{bmatrix} &= 0 \\
\sum_{k=1}^{N_k} \begin{bmatrix} \mathrm{R}_\theta \mathrm{H}_{q,k} \mathrm{R}_\theta{}^{-1} & \mathrm{R}_\theta \mathrm{H}_{q,k} \mathrm{S} q_{xy,k} \\ q_{xy,k}^\top \mathrm{S} \mathrm{H}_{q,k} \mathrm{R}_\theta{}^{-1} & q_{xy,k}^\top \mathrm{S} \mathrm{H}_{q,k} \mathrm{S} q_{xy,k} \end{bmatrix} \begin{bmatrix} \dot{p}_{xy,0} \\ \dot{\theta} \end{bmatrix} &= \sum_{k=1}^{N_k} \begin{bmatrix} \mathrm{R}_\theta \mathrm{H}_{q,k} \dot{q}_{xy,k} \\ q_{xy,k}^\top \mathrm{S} \mathrm{H}_{q,k} \dot{q}_{xy,k} \end{bmatrix} \\
\sum_{k=1}^{N_k} \begin{bmatrix} \mathrm{H}_{q,k} & \mathrm{H}_{q,k} \mathrm{S} q_{xy,k} \\ q_{xy,k}^\top \mathrm{S} \mathrm{H}_{q,k} & q_{xy,k}^\top \mathrm{S} \mathrm{H}_{q,k} \mathrm{S} q_{xy,k} \end{bmatrix} \begin{bmatrix} \mathrm{R}_\theta{}^{-1} \dot{p}_{xy,0} \\ \dot{\theta} \end{bmatrix} &= \sum_{k=1}^{N_k} \begin{bmatrix} \mathrm{H}_{q,k} \dot{q}_{xy,k} \\ q_{xy,k}^\top \mathrm{S} \mathrm{H}_{q,k} \dot{q}_{xy,k} \end{bmatrix}
\end{aligned}
$$

Hence, body velocity $\dot{p}_{xy,0}$, $\dot{\theta}$ can still be solved linearly with respect to shape changing velocity $\dot{q}_{xy,k}$, giving a general form:

$$
\begin{aligned}
\begin{bmatrix} \mathrm{R}_\theta{}^{-1} \dot{p}_{xy,0} \\ \dot{\theta} \end{bmatrix} &= \sum_{k=1}^{N_k} \left(\sum_{k=1}^{N_k} \begin{bmatrix} \mathrm{H}_{q,k} & \mathrm{H}_{q,k} \mathrm{S} q_{xy,k} \\ q_{xy,k}^\top \mathrm{S} \mathrm{H}_{q,k} & q_{xy,k}^\top \mathrm{S} \mathrm{H}_{q,k} \mathrm{S} q_{xy,k} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathrm{H}_{q,k} \\ q_{xy,k}^\top \mathrm{S} \mathrm{H}_{q,k} \end{bmatrix} \dot{q}_{xy,k} \\
&=: \sum_{k=1}^{N_k} A_k(q) \dot{q}_{xy,k} \\
&= A(q) \dot{q} \qquad (3.20)
\end{aligned}
$$

where $A(q)$ is constructed by stacking the block of $A_k(q)$ if that foot is contacting, and zeros otherwise; $\dot{q}$ is the stack of xy foot velocities in body frame. The $A_\cdot(q)$ matrices are the kinematic term (mechanical connection) in the reconstruction equation of geometric mechanics.

3.5 Experimental validation

To verify our model's accuracy in practice, we compared our model predictions to ground truth motion capture measurements of three different multilegged robots. We tested our model on the BigAnt robot with force-torque sensors measuring the ground contact wrenches. We compare the robot motion prediction, supporting force prediction, and 2D friction prediction of friction and Coulomb friction model with the robot executing two different gaits: tripod gait and metachronal gait. We further verified the motion prediction of the model against Multipod robots with 6-12 spring legs running undulation gait, and a commercial quadruped, Ghost Robotics Spirit with the company-provided onboard controller.

The marker-based motion capture system used in all experiments consisted of 10 Qualisys Oqus-310+ cameras, running at 100 fps with QTM 2.17 build 4000 software.

3.5.1 BigANT with force-torque sensors

3.5.1.1 Robot and measurement setup

To experimentally verify our algorithm, we built a version of the BigANT robot (figure 3.5) with a 6-DoF ATI Gamma force-torque sensor attached to each leg, and used ATI's wireless F/T sensor system to communicate the measurements to the controlling host computer at 100 Hz. We calibrated the sensors to report the contact forces at the feet. This calibration process is documented in the next chapter of this thesis.

BigANT is a hexpedal robot that has only one motor (Robotis Dynamixel MX106) per leg. The leg drive-train is a four-bar linkage designed to provide advantageous foot clearance and gearing [8, Chapter 2.2]. We manufactured the legs from foamcore (Elmer's Products Inc. 3/8" foam board) and fiber reinforced tape (3M Scotch #8959) using the "plates and reinforced flexures" (PARF) technique of [56]. We laser-cut the base plate for the BigANT chassis from a 1/4" ABS plastic plate (Mc.Master-Carr).

We recorded the robot motions with the motion tracking system. This allowed us to measure the position and orientation of a robot body frame from markers we attached to the robot's chassis, and take the foot positions (q_j) relative to that body frame. We obtained foot velocities $\dot{q}_j$ by differentiating q_j using a Savitzky-Golay filter (from `scipy.signal.savgol_filter`, 2nd order, with window size 25).

3.5.1.2 Parameter fitting

The remaining inputs to the algorithm were not so easy to determine. Because our model is quasi-static, the mass plays no direct role, except for its appearance in Mg as the sum total of normal forces at the feet. The force and moment balance equations remain unchanged regardless of the units selected for force, and these affect only Mg, the stiffnesses K, and the friction coefficients H. We therefore chose $Mg = 1$. Using marker positions, we estimated the robot body's height, pitch and roll according to §3.3.

We estimated the spring constants K and two anisotropic friction model coefficients per leg (μ_k and $w_{xy,k}$ of eqn. (3.19)) using least-squares minimization of a suitable loss function: we minimized the L2-norm difference between modeled and measured $F_{z,}$ distribution among contacting legs, while adding the variability of K among legs as a regularization penalty. We assumed anisotropic friction coefficients H_k (see eqn. (3.19)), and inferred the parameters μ_k and $w_{xy,k}$ for each leg by minimizing L2-norm error between measured forces and forces calculated from slipping velocity measured by motion capture. We used `scipy.optimize.least_squares` for both of these parameter estimation minimizations.

In total, we fitted 18 constant model parameters to predict a time series of six 3-dimensional leg forces and 6-DoF body velocity measurements, thus there is little risk of over-fitting.

3.5.1.3 Coulomb friction solver

We solved for the Coulomb friction force-balance using `scipy.optimize.root` with the LM algorithm, hot-starting with the solution of the previous time-step as an initial guess. Because Coulomb friction is non-smooth, we employed a sequence of smooth approximations to the solution starting from $\varepsilon = 10^{-5}$ (from eqn. (3.14)), using each solution as an initial condition for another solution with smaller ε until the relative change in L2-norm of two consecutive solutions differed less than 10^{-3} – a threshold less than 5% of the median ground speed measured. In the very rare cases (0.12% of the BigANT tripod gait dataset) where the Coulomb friction solver failed to converge, we used the ground truth velocity as the initial condition to obtain the Coulomb friction solution; this process resolved the remaining cases.

3.5.1.4 BigANT using tripod gait

We first ran the robot through an alternating tripod gait driven with the "Buehler clock" of [6] and the steering strategy described in [57]. We collected 21 tripod gait trials with the robot running at 0.1Hz (dataset at [31]), with 4-5 cycles in each trial, and a total of 102082 frames consisting of 84 ± 1 cycles. The motions of the shaft angles were scheduled to have a slow ground contact phase and a fast leg reset phase. We show in figure 3.5 a comparison of forces and kinematics modeled by our multi-contact algorithm with viscous-Coulomb friction, our algorithm with classical Coulomb friction, and the experimental measurements. We integrated body velocity and showed the robot trajectory in figure 3.5.

Because our physical modeling assumptions only define contact forces up to a positive scale factor, we chose a single positive scalar $\sigma(t)$ for every time step, such that the loss function

$$\sigma := \arg\min_{c} \sum_{k} (c|\hat{F}_k| - |F_k|)^2$$

between the 12 dimensional prediction $\hat{F}$ and the measured horizontal forces F was minimized.

We reported prediction error statistics in figure 3.7. The run time for viscous-Coulomb friction was 0.19 (0.18, 0.24) [ms/frame; mean, 1st and 3rd quantile]. When running a single approximation with the choice of $\epsilon = 10^{-5}$, the Coulomb friction solver took 3.7 (3.1, 3.9) [same]. For a full set of iterations to convergence, Coulomb friction took 10.4 (3.25,15.0) [same], about $\times 54$ slower than our viscous-Coulomb ansatz based simulation.

To test the robustness of our modeling approach, we measured the predictions' sensitivity to the

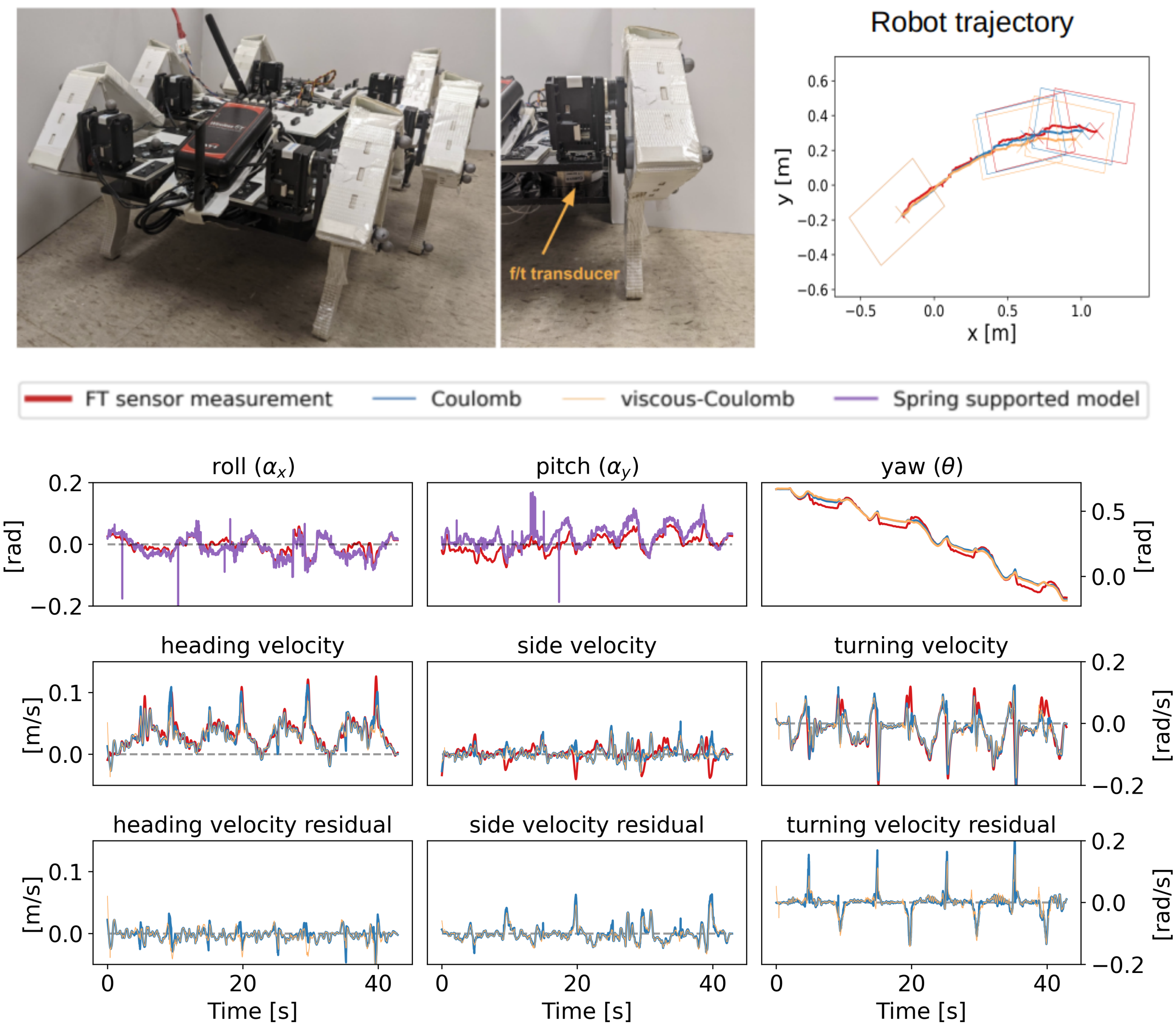

Figure 3.5: A tripod gait trajectory of BigANT. We plotted the trajectory of the BigANT robot (top row) measured from motion capture (red). We plotted roll, pitch, and yaw angles (second row, left to right), and body velocity components along the robot axis, across the robot, and rotational (third row, left to right). We plotted body velocity model residual (fourth row). We used motion capture data as ground truth for kinematics (wide red lines). The pitch and roll are the same for both friction models and arise from the spring support model (purple lines), but other outputs are different for Coulomb friction (blue lines) and our viscous-Coulomb friction (orange lines). We indicated the body location obtained from motion tracking and by integrating the body velocity predictions (rectangles in "Robot Trajectory"). We further illustrated the motion by indicating the location of the robot body frame origin (crosses) at the beginning, halfway point, and end of motion (same sub-plot).

model parameters – the spring constants K and the anisotropic friction coefficients μ_k and $w_{xy,k}$ in eqn. (3.19). We compared the prediction using naive parameters with the fitted parameters as

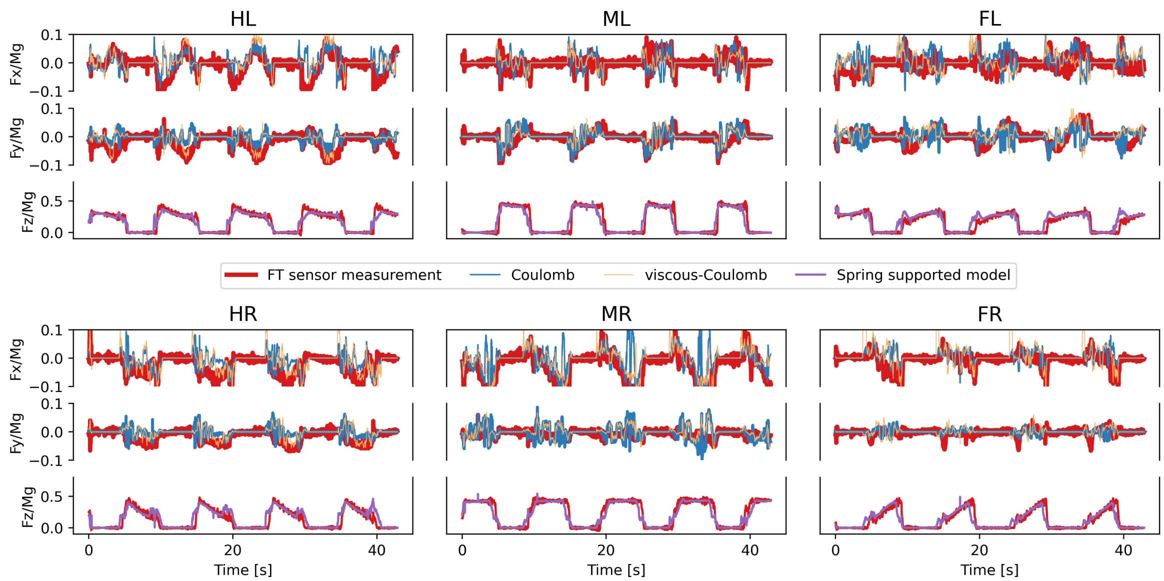

Figure 3.6: Individual leg contact forces measurement and prediction for BigANT robot with tripod gait. We plotted the ground contact force in x, y-axis, and supporting force ratio in z-axis of each individual leg. We used force torque sensor measurements as ground truth for forces (red). We plotted the estimated z-axis force ratio (purple), estimated xy-axis friction forces (viscous-Coulomb: orange, Coulomb friction: blue). (leg names are: HL hind left, ML mid left, FL front left, HR hind right, MR mid right, FR front right)

described in §3.5.1.2. We used the same spring constants $K = 10$, and only isotropic friction coefficients $\mu = 1$ for all legs. We chose the spring constant magnitude so that the model could lift its feet to perform the appropriate tripod gait. The body velocity prediction root-mean-square error (RMSE) in the heading direction increased by 14%, in the side direction decreased by 5.5%, and turning increased by 1.5%. Roll prediction RMSE decreased by 22%, pitch RMSE increased by 24%. Friction force prediction RMSE among all legs increased by 4% and 3% in heading and side direction respectively. The normal force prediction RMSE among all legs increased by 2%.

3.5.1.5 BigANT: metachronal gait

We wanted to further study why the viscous-Coulomb friction model gave similar body velocity and force predictions to those of the classical Coulomb friction model. Since non-slip motions provide little insight into the question of which friction force model to use, we developed a metachronal gait with exacerbated multi-legged slipping events. Each foot contacted the ground for 2/3 of a cycle leading to four feet, two from each side, being in contact with the ground at any time (see figure 3.8). To ensure that feet slip, we needed to ensure that the distances between contacting feet change while in stance. We facilitated this by ensuring that the contact feet have vastly incompatible

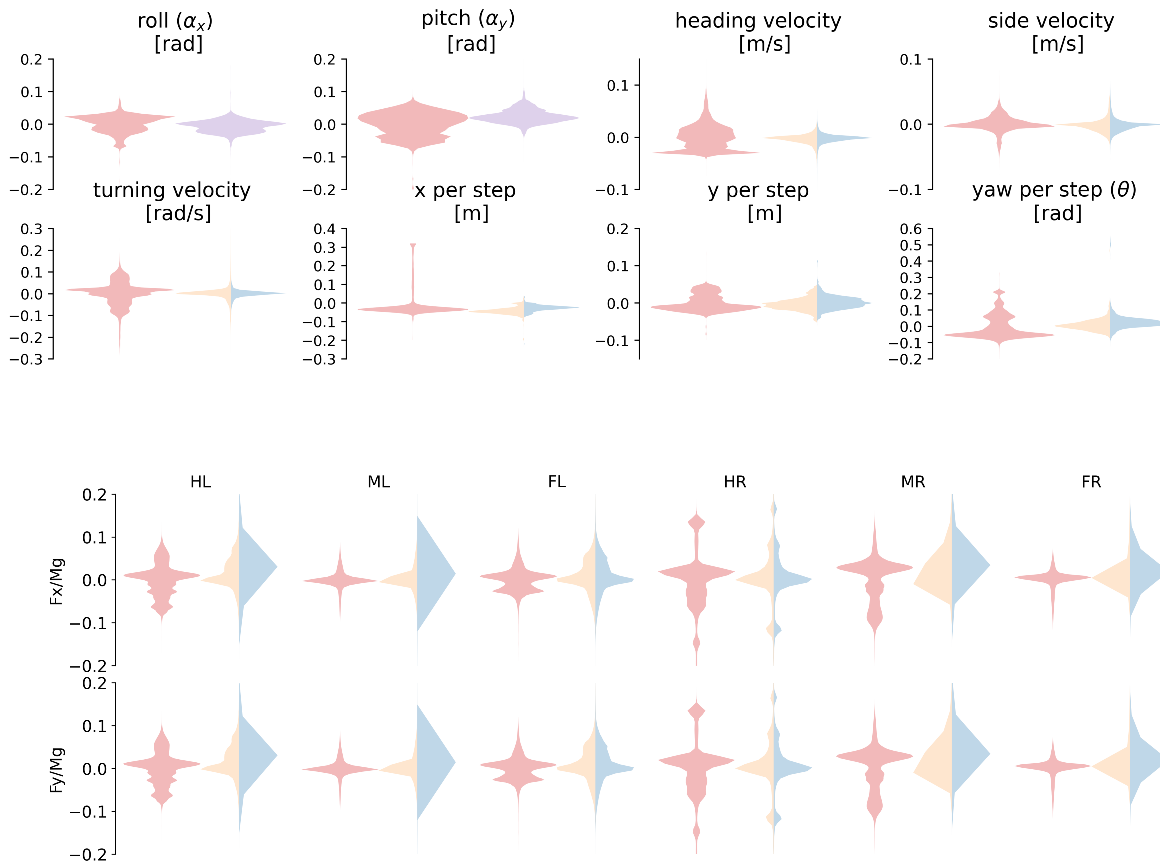

Figure 3.7: Prediction error distributions for BigANT robot with tripod gait. We plotted the distribution of measurement residuals from the mean (red) to compare with residuals of the predictions of the spring supported model (purple), or residuals of predictions from viscous-Coulomb (orange) and Coulomb friction (blue).

velocities by choosing the shaft angle to be a cubic function of time during stance. We collected 12 metachronal slipping gait trials, with the robot moving forward 4-6 cycles in each. In total, the data consisted of 43934 frames and 60 ± 1 cycles. The resulting gait produced much more slipping than the tripod gait, with slipping velocities ranging in $(-51.8, 111.8)$[mm/s; 5% , 95% percentiles].

To determine whether viscous-Coulomb or classical Coulomb friction was indicated in these data, we examined the force measurements from the slipping gait. Plotting $F_{x,j}/F_{z,j}$ against $v_{x,j}$ (see figure 3.8[B]) shows the expected structure for classical Coulomb friction, namely a step function.

3.5.2 Demonstration with other legged systems

To test whether the proposed model generalizes to other legged systems, we further examined using our model on Multipod robots with 6-12 leaf-spring legs and an undulation gait, and on the commercially available quadruped Ghost Robotics Spirit 40.

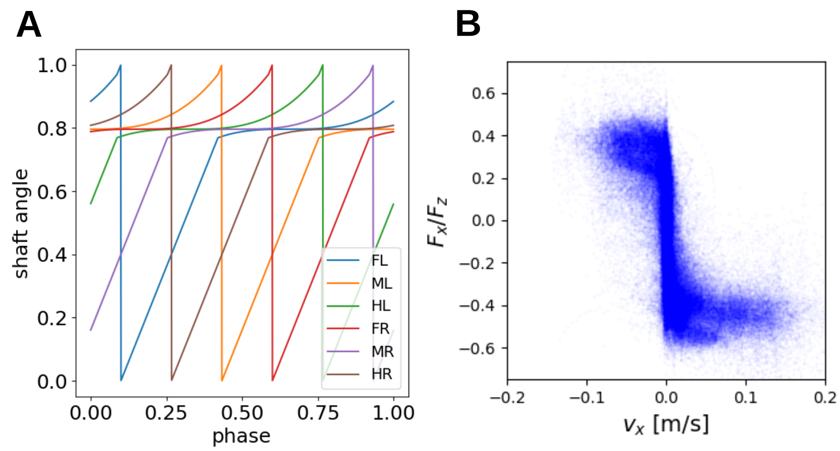

Figure 3.8: We plotted: (A) Metachronal gait phase vs. motor shaft angle for all six legs. (B) Magnitude of slipping velocity vs. magnitude of planar force divided by normal force, overlaying points from all six feet.

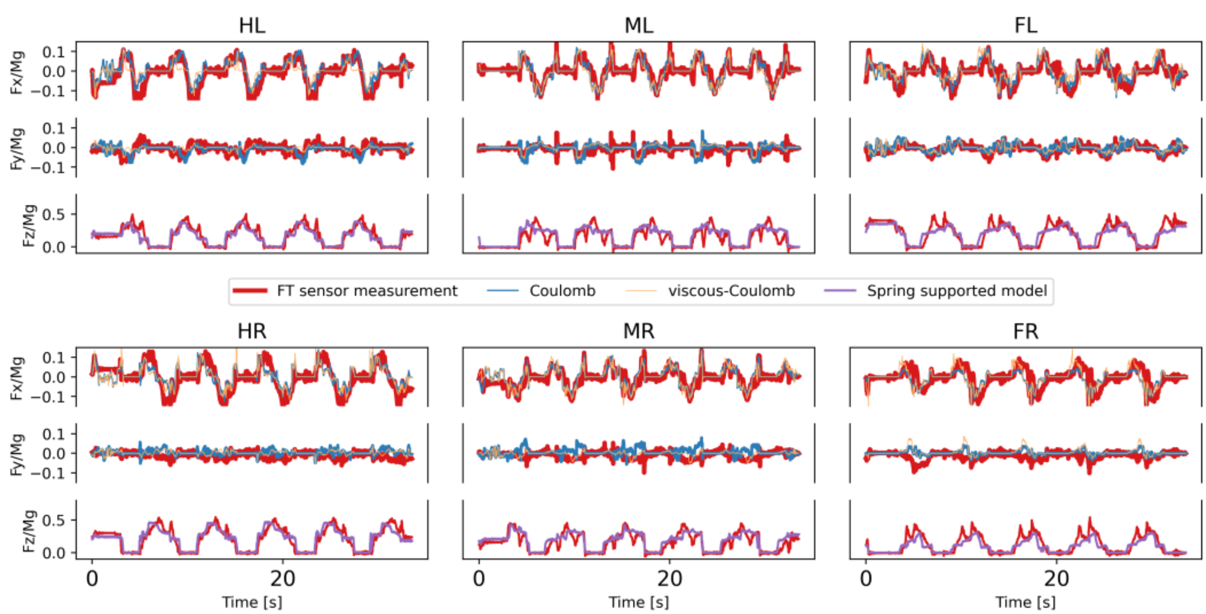

Figure 3.9: Individual leg contact forces measurement and prediction for BigANT robot with metachronal gait. We plotted the ground contact force in x, y-axis, and supporting force ratio in z-axis of each individual leg. We used force torque sensor measurements as ground truth for forces (red). We plotted the estimated z-axis force ratio (purple), estimated xy-axis friction forces (viscous-Coulomb: orange, Coulomb friction: blue). (leg names are: HL hind left, ML mid left, FL front left, HR hind right, MR mid right, FR front right)

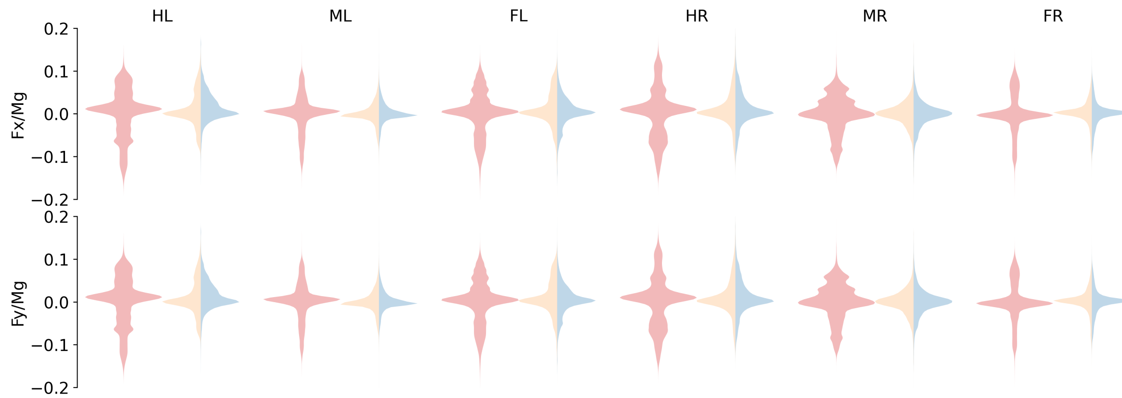

Figure 3.10: Planar forces prediction error distributions for BigANT robot with metachronal gait. We plotted the planar forces with mean subtracted (red) and model prediction errors for planar forces (viscous-Coulomb: orange, Coulomb:blue).

3.5.2.1 Multipods with 6, 8, 10, and 12 legs

We used our lab's publicly available Multipod dataset [58, 59, 60] used in [30]. Each contralateral pair of legs in a multipod has two DoF – yaw and roll – and the roll (vertical) motion is further compounded with the spring flexing of the leg itself (see figure 3.11). To model the body motion of unloaded spring legs, we computed the location of each foot relative to the rigid segment to which its spring was connected based on those motion tracking frames in which the leg was unloaded. We then extrapolated this unloaded position to the frames where the leg was loaded.

In figure 3.11 we used a slice of these data with the robot running at frequency 0.3Hz and phase offset 1.35π to demonstrate our algorithm. We assumed the mass of the robot is linear in the number of legs – an explicit design feature of these robots – and set $Mg = N$. We used $K = 1$ as the spring constant and isotropic friction model $\mu_k = 1$ on all legs.

3.5.2.2 Ghost Robotics Spirit

Our physical modeling approach was built upon the assumption that friction dissipates the robot's body momentum quickly in comparison to the time scale of gait motions. We intentionally selected a commercial quadruped, the Ghost Robotics Spirit, where this assumption breaks down, to test how well the connection-based model could approximate the motion of such a quadruped. We used the Spirit 40 v1.0 Proto Q-UGV Robot from Ghost Robotics, operated through a Samsung Galaxy A50 with onboard firmware version 0.12.6. We collected 921 frames comprising about 9 cycles of motion (see figure 3.12). Because our model has no inertia, it tends to produce spurious

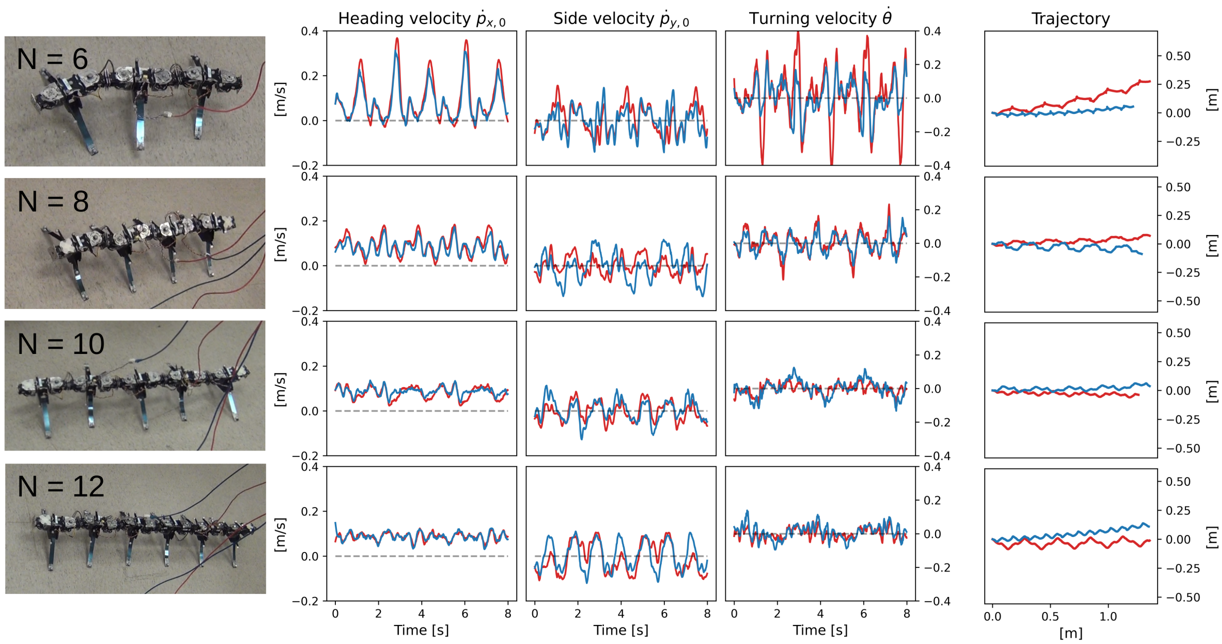

Figure 3.11: 6 and 12 legged Multipod gait. Here we plotted the body velocity and predicted body velocity at 0.3Hz and $1.35\,\pi$ phase offset. We show Multipods with 6 legs (top) and 12 legs (bottom). We plotted the estimated (blue) velocity (middle) and trajectory (right), for comparison with the motion tracking (red). Side velocity plots have the same unit and scale as the heading velocity plots.

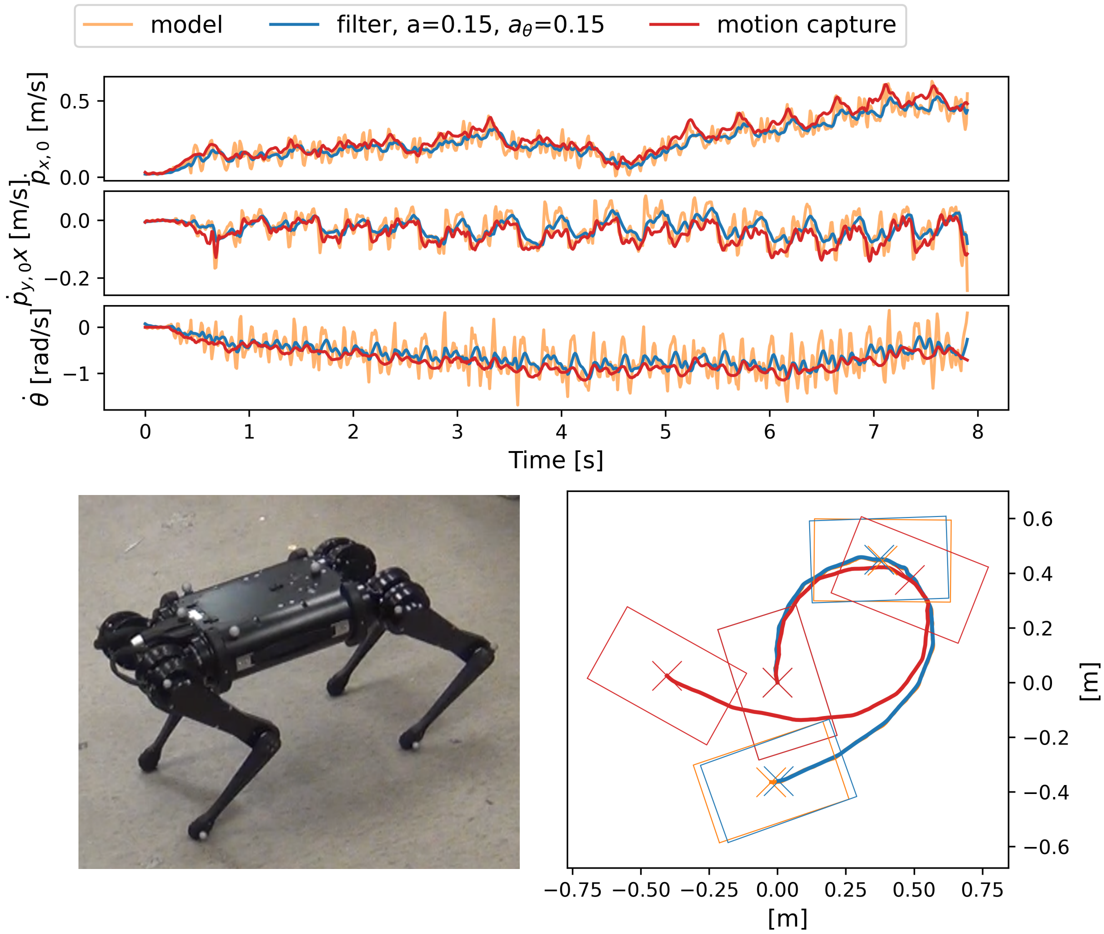

Figure 3.12: Modeling of a Ghost Robotics Spirit 40. We plotted the estimated velocity (top three rows, in orange) of the robot (left, bottom row), filtered estimation (blue) and motion tracking (red) of the absolute motion (right, bottom row).

high-frequency changes in its predictions. To obtain a more realistic time series, we added a simple model of robot inertia in the form of a first-order IIR lowpass filter $y_n = \gamma y_{n-1} + (1-\gamma)x_n$, where x_n is our raw model prediction and y_n is the filtered prediction. We manually selected $\gamma = 0.15$ to bring the power spectral density (PSD, computed using `scipy.signal.welch`) of the estimated body velocities close to that of the motion tracking derived velocities.

3.5.3 Model runtime analysis

We compared the computation speed between our algorithm using the viscous-Coulomb friction model and a widely-used physics simulation engine MuJoCo [23, v2.2.1]. MuJoCo performs competitively with other physics simulation engines (see [61], but also [62]), and by restricting itself to convex meshes and relaxing its contact condition, it is often considered for problems with many independent contacts.

Since our focus is on multi-legged contacts, our models consisted of a round disk with 3 to 50 legs equally spaced on its circumference. We gave each leg two rotational DOFs, a vertical translation DoF, and limited leg motions so that their workspaces did not overlap. We tested the execution time of both MuJoCo and our algorithm at 1000 randomly chosen poses and velocities for each number of legs, and re-normalized the running time by dividing by the median execution time of the 3-legged cases, to reveal how each simulation approach scaled with the number of legs (see figure 3.13). While both algorithms reveal an increase in execution times, our algorithm slows down by less than a factor of 3 with 50 legs, compared with a factor of 13 for MuJoCo. This suggests that an optimized implementation of our algorithm could be used for multi-legged motion planning for any practical number of contacts.

Because we are using an inertia-free model of physics in the form of a local connection, the body velocity at any instant is only a function of the shape change and shape velocity at that instant. Hence, in a homogeneous environment, all time-steps of a motion plan can be computed in parallel. To demonstrate the performance gains, we simulated 10,000 random poses and velocities of a hexapod robot. We used $P = 1, \cdots, 4$ processors to compute the body velocity matrices in parallel, then integrated them in a single linear process (note: this over-estimates the parallelization overhead, since the product of N matrices can be parallelized to take $\log_2 N$ time, but was linear here). In figure 3.13(b) we show that the algorithm parallelizes well, with the overhead at four processors falling below 1.5, i.e. a net speedup of $4/1.5$.

3.5.4 Model accuracy comparison with Mujoco

We compared our model accuracy with Mujoco (v3.1.4). We used Newton's algorithm with maximum 100 iterations for the compiler option. We modeled the robot with a plate chassis having `diaginertia='130000 430000 300000' mass='818'`, and almost mass-less rod legs with `mass='0.1'`. We used three linear position actuators per leg to control the foot position. We chose the parameters of `damping='10'` for all joints, and `kp='100'` for actuators along xy, and `kp='1000'` for actuators along z. The actuator parameters were chosen so that the foot trajectory executed by Mujoco simulation has RSME less than 1cm on each foot, when controlled to be the experimental measurements. To the best of our parameter tuning effort, we did not see MuJoCo out perform our proposed model in model accuracy. An example robot trajectory comparison was plotted in figure 3.14.

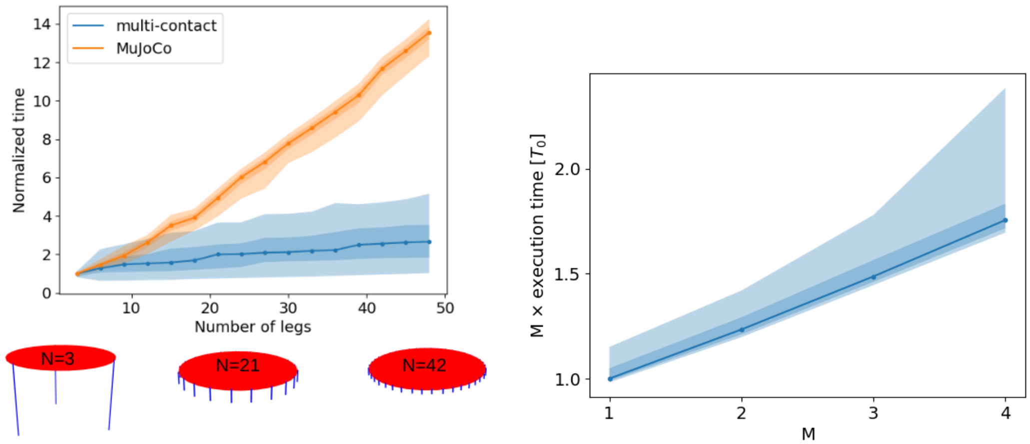

Figure 3.13: (a) Plot of the normalized run time of the multi-contact algorithm and MuJoCo simulation versus the number of legs. We plotted the distribution of time-step computation times on 1000 randomly initialized configurations for each number of legs from 3 to 50. Plot indicates distribution percentiles 2.5 to 97.5 (lightly shaded); 25 to 75 (shaded); and median (dotted line). The execution times are normalized relative to the median execution time of each simulation on the 3 leg case. The robot configurations consisted of a disk with N equally spaced legs on the rim as illustrated by examples with $N = 3$, 21 and 42. (b) Plot of parallelization overhead splitting the algorithm over M threads. The overhead is execution time times M the number of threads, in units of the median execution time on a single thread. Perfectly parallelizable workloads give 1 whereas unparalellizable workloads give M. We plotted the workload distributions at $M = 1 \ldots 4$ for a hexapod, running 100 randomized trajectories each 10000 time-steps long (ribbon with the same quantile as in (a)).

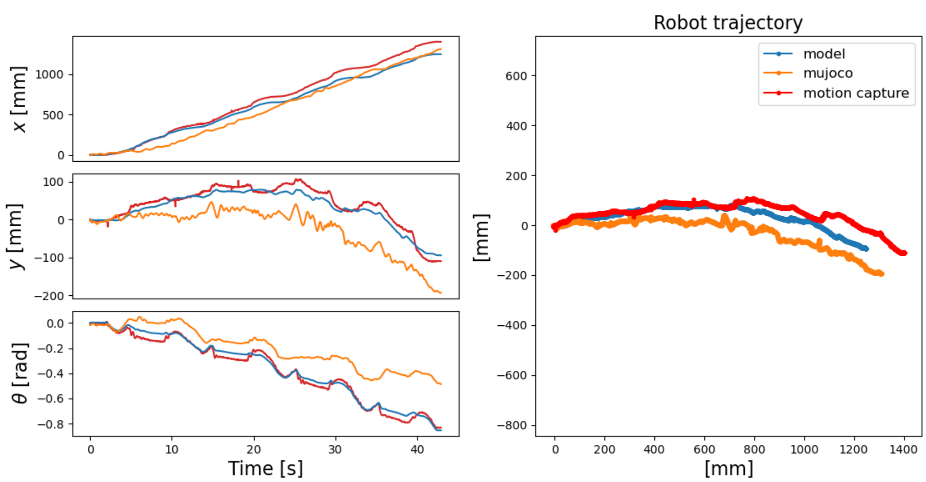

Figure 3.14: We plotted the robot position x, y and orientation θ with respect to time (left) from our proposed model (blue), Mujoco (orange) and motion capture (red). We showed the robot integrated trajectory (right).

3.6 Conclusions and discussions

Multi-legged robots (with six or more legs) are not widely studied in the robotics community. One reason might be that the complexity of modeling the multi-contact ground interaction constrains both motion planning and simulations for design.

Motivated by a previous discovery [30] – that multi-legged robots move as if they are governed by a local connection, i.e. quasi-statically – we developed a simplified ground-interaction model and validated it experimentally. Our algorithm consists of simulating a spring support body in a small-angle approximation for pitch and roll to obtain the vertical foot loadings. We then introduced the viscous-Coulomb ansatz to replace classical Coulomb friction in generating the horizontal forces to produce a linear set of equations that can be solved to give rise to the local connection.

3.6.1 Computational speed

Our experimental verification demonstrated that while the actual contact forces were, as expected, governed by classical Coulomb friction, our viscous-Coulomb friction model gave equally good predictions of both contact forces and body velocities, while computing 50 times faster for a hexapod. Our algorithm scales to large numbers of contacts with virtually no change in execution time, and parallelizes with very low overhead. Although all our computation was done offline, the computations, even when done in Python, ran about 50 times faster than in real time. Thus it is quite clear that our model can easily be used in online form.

Our model does not sacrifice accuracy for computation speed. We compared the robot trajectory prediction from our model with MuJoCo in figure 3.14. To our best effort in tuning the MuJoCo parameters, we showed the MuJoCo prediction did no better than our model in practice. Therefore, the proposed model, a much faster model with a small sim-to-real gap, could benefit the reinforcement learning community in getting optimal policies on multi-legged robots.

3.6.2 Comparing viscous-Coulomb friction model to Coulomb friction model

To understand how a system governed by sliding Coulomb friction can be modeled by a viscous-Coulomb friction model, one may examine and compare the relative error of a viscous friction model to that of the "true" Coulomb friction. Because both models are isotropic, we can assume without loss of generality that the velocity is in the x direction. Because both models are homogeneous, we can assume without loss of generality that the speed is 1. What remains is to study the relative error of predicting the Coulomb friction force for contact velocities close to $(1, 0)$ and the prediction obtained by using the viscous drag model instead. In figure 3.15, we present the contours of relative error when using a single viscous friction model instead of Coulomb friction over the specified

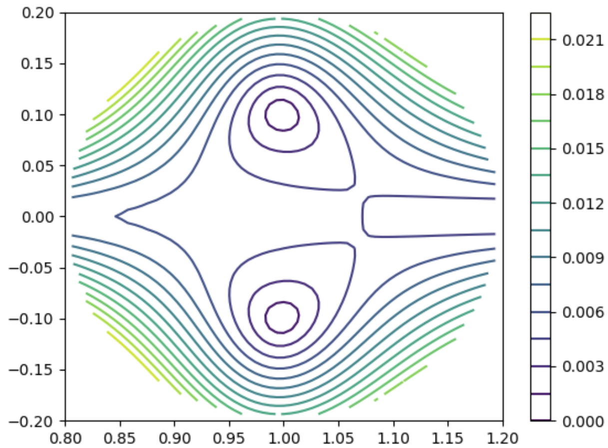

Figure 3.15: Contour of error between viscous-Coulomb approximation to Coulomb friction around equilibrium velocity.

range of velocities. The plot demonstrates that with $|\delta v| < 0.2|v|$, the viscous-Coulomb force prediction for velocity $v + \delta v$ will be within 2% of the classical Coulomb friction force prediction. This suggests that viscous-Coulombfriction is a good approximation of Coulomb friction when the sliding velocities do not fluctuate more than 20%.

The linearity between slipping velocity and friction forces was observed as an average relationship in the numerical simulations of [32] and in the experiments of [54]. In [54], the authors study the slipping locomotion of sprawled posture robots with many legs and biological myriapods, and discover a linear force-velocity relationship near steady state velocity. The authors assume the net force on an individual stance foot is equal to 0, the slipping is predominantly in the lateral direction, and the supporting force is constant. In our work, we relax these assumptions by solving for the force-torque balance at the robot body level, allowing internal forces among the contacting legs; we do not restrict slipping directions, and we allow supporting forces to vary through the spring loaded model.

We are thus left with the conclusion that a viscous-Coulomb ansatz model for friction produces very similar predictions to those produced by the classical, tribologically accurate Coulomb friction model. Comparing the motion predictions obtained from both models, they are far more similar to each other than either is to the measured motion, suggesting that the dominant error in these models was not the use of an incorrect friction model. However, the viscous-Coulomb model, in the context of our multi-contact algorithm provides a significant performance boost. It is faster to compute; it scales better with the number of contacts; and it is easier to parallelize.

From the perspective of physics, that our ansatz produces motion plans as accurate as those produced by Coulomb friction but also provably produces a local connection and principally kinematic motion in the geometric mechanics sense is further justification for the observation of [30] that local connection models are a framework that includes multi-legged locomotion. While the local connection models of [30] were data-driven, here we have shown that such models can be obtained using a principled modeling approach.

3.6.3 Future works

The algorithm we presented here provides merely a starting point – it is a means for rapidly and accurately estimating multi-contact robot-environment interactions. Such estimates are building blocks for motion planning, model predictive control, design optimization, and many other potential applications. The algorithm itself can be extended to include contacts with non-flat ground. For example, if the ground can be reasonably modeled as horizontal surfaces at various elevations, such as for staircases, we could encompass this in our model with a leg length offset as a function of the foot position in the world frame.

Modern state observers for quadruped robots use the fusion of proprioceptive information and inertial measurement to estimate state while slipping [63]. These have recently been extended with learning to improve contact estimation and odometry [64]. Like any recursive estimator, such estimators alternate between measurement steps and system evolution steps. It would be interesting to explore how to integrate our model as the system evolution operation in such a scheme. Additionally, the various quantities we estimated by fitting could be converted to slowly varying online estimates, producing an adaptively tuned controller.

Because our proposed model provides a linear relationship between shape space velocity and body velocity, it provides a first-order representation of the dynamics as an affine control system, which could be useful in simplifying the controller design process.

Our model could potentially be used as an extension of the existing models for snake slithering. First, one might discretize the snake into a 3D shape with appropriately chosen body velocities and vertical stiffnesses. The simulated snake would then relax into the surface, potentially contacting with only a part of its body, and generate a local connection between its body motion and its slithering speed. Such an approach might provide models for sidewinding and other fully 3D snake behaviors.

We hope that our advances will stimulate the adoption of multi-legged robots in field robotics, provide reliable and adaptable bio-inspired locomotion platforms, and, more generally, enable the modeling of multi-contact mechanics problems.

CHAPTER 4

In-situ calibration of six-axis force/torque transducers on legged robot

4.1 Introduction

Force/torque (F/T) transducers are commonly used in robotics applications to sense the interaction between robots and their environment [65]. The simultaneous and accurate measurement of multiple concurrent contact forces through, e.g. a robot's legs, has proven to be a challenging calibration problem, for which we present one solution in this chapter. This solution proved effective for performing ground contact measurement on our 8.18 kg, 73 cm long hexapedal robot BigANT.

F/T transducers are used in robotics surgery [66], manipulation [67, 68], robot locomotion [69, 70, 71], etc. Knowing contact forces can allow a robot to sense its surroundings more fully, with ensuing benefits to both planning and control. For legged robots which move by making and breaking contacts with the ground to produce propulsion and load bearing, F/T transducers can measure how the load is distributed among legs. The measurements can reveal how much propulsion each leg produces, what slipping might occur, and thereby inform modeling and control. Such F/T measurements have been used in contact detection [69], gait modeling [70], state estimation [72], terrain classification [71], maintaining stability [73], and other applications in legged robotics.

Our work was motivated by the goal of understanding the interplay of contact forces while a multi-legged robot moves with feet slipping on the ground. We are particularly interested in multi-legged robots with low degree of freedom (DoF) legs [74, 75]. The mass of such robots is typically concentrated in the body, allowing fast swinging legs, lower leg collision forces, and smaller energy costs for leg impacts. Our previous work showed that slipping was an essential part in expressing the high maneuverability of these robots [76], but due to a lack of precise contact force measurements our mechanistic understading of these complex slipping motions was incomplete. In the previous chapter, we managed to reliably model the body velocity resulting from a multi-legged slipping motion, by constructing a "local connection model", by assuming a viscous-like friction interaction with the ground instead of the expected Coulomb friction. Our goal was to understand what interactions contributed to the unexpectedly good predictive power of the “wrong” friction model, and what are the actual contact forces involved in the motions.

We built a version of the BigANT robot [8] with a 6-DoF F/T transducer at the hip of each leg. This, we believed, would enable us to measure the wrench applied by each leg to the robot chassis, and since the shape of the leg and its point of contact were known from motion tracking, to compute the contact force that produced each leg’s wrench. We chose to install the F/T transducers at the hips, mounted directly on the robot’s chassis to maintain the lightweight, fast-swinging nature of the legs. Additionally, the commercial transducers we found for the range of wrenches expected were both heavy and fragile to impacts, making it all but infeasible to attach them directly to the tip of a foot. The wrench each transducer measured comprises the ground contact wrench, added to the wrench generated by gravity acting on the (albeit lightweight) leg. Our calibration procedure needed to decouple these two terms to isolate the ground reaction wrench we wanted to measure.

Few have measured multi-legged contact wrenches simultaneously at all active contacts. One example is in [50] wherein the authors measured 3D ground reaction forces for a version of the hexapod RHex. The authors did not provide details on characterization and analysis of measurement error, but did provide 2N as the maximum error. The authors of [77] performed in-situ calibration with F/T transducers mounted on the shoulders, hips and feet of a bipedal robot. They used the model from their Computer Aided Manufacturing (CAD) design file to extract the center of mass (CoM) of each segment and estimate the torque resulted from the gravity of each segment.

Below we present a calibration procedure enabling simultaneous measurement of all foot contact wrenches of our multi-legged robot using six 6-DoF F/T transducers, one at the hip of each leg. We start with introducing the experimental setup in §4.1.1, followed by difficulties we met when using the transducers naively according to the manufacturer’s guidlines in §4.1.2.1. We present the problem statement including notations and F/T measurement model in this calibration work in §4.2, and follow with by our calibration method in §4.3. We then show our validation of the calibration results in §4.4 and end with a discussion in §4.5.

4.1.1 Experimental system: robot and sensors

Our goal was to measure the ground contact wrenches with respect to a “floating base” frame attached to the robot chassis. The robot we used was a version of the BigANT hexapod [8] (figure 4.1) constructed using PARF methodology [75]. The z axis of the frame was opposite to direction to gravity, and the x axis perpendicular to z and parallel to left-right symmetry axis of the robot. The origin of the floating base frame was taken to be close to the centroid of the robot chassis.

Because of all the instrumentation planned for this robot, it was far heavier than previous BigANT

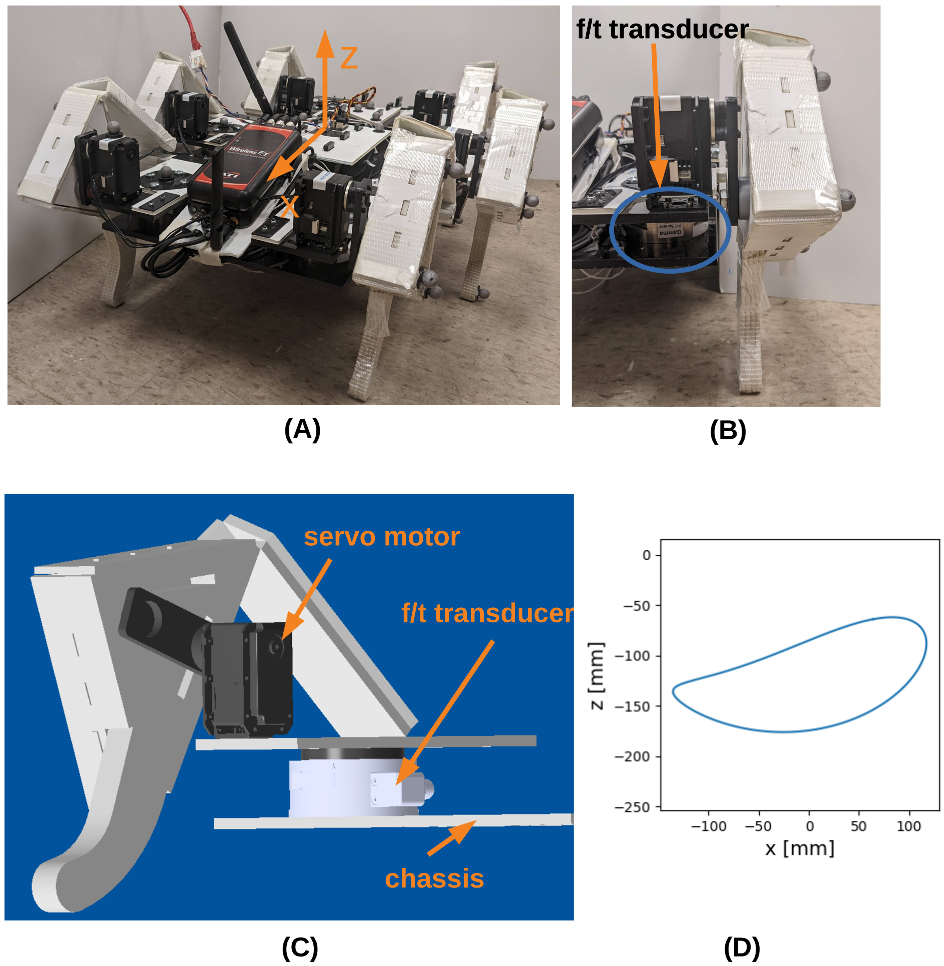

Figure 4.1: (A) BigANT equipped with six 6-axis force/torque transducers. (B) Zoomed-in view of one leg with F/T transducers. (C) Schematic drawing of front right leg module. (D) 1-DoF toe tip trajectory in robot body frame xz-plane.

variants. To ensure better rigidity under these loads in comparison other BigANT variants we used a 1/4 inch Acrylonitrile Butadiene Styrene (ABS) sheet for the chassis base-plate. Connecting each leg to the robot chassis, we installed an ATI Net Force/Torque Gamma transducer, which measured all six components of force and torque at 100 Hz and reported the results over WiFi. As in other BigANT robots, we actuated each leg with a position controlling servo motor module (Robotis Dynamixel MX106). We daisy chained the modules as they were designed to be used, and communicated with them over the RS-485 serial bus carried over two wires of this daisy chain. The total weight of the robot was 8.18 kg, measured by a digital hanging scale. Along with the F/T data, we also recorded robot kinematic data using a reflective marker motion tracking system (10 Qualisys Oqus-310+ cameras at 100 fps, 12 markers on the chassis and 4 markers per the distal rigid part of each leg).

Each leg of BigANT is driven by a four-bar mechanism producing a shaped toe trajectory (figure 4.1) with a variable gearing ratio. Because the legs were effectively rigid 1-DoF mechanisms (as verified using motion tracking), the shape of each leg changed through a one-parameter family of shapes entirely controlled by the motor shaft angle of the motor driving that leg. Consequently the location of the CoM of each leg (with respect to the chassis frame) was also solely a function of that leg's shaft angle. Therefore the gravity torque introduced by each leg on its hip F/T transducer was a function of the direction of gravity in the chassis frame (shared among legs), and the shaft angle of the individual leg.

4.1.2 Challenges in accurate ground contact wrench measurements

4.1.2.1 Pre-experimental F/T transducer measurement validations

As a sanity check prior to collecting motion data, we collected measurements with the robot standing statically at different poses. Because acceleration is zero, the total wrench on the robot body should be equal and opposite to gravity acting on the robot body through the body's CoM, i.e. when accounting for gravity, the sum of forces on the robot body should be zero. The ATI use manual [78] suggests that the transducers may suffer from a constant bias wrench which needs to be subtracted. We used the average transducer readings obtained with all feet in the air rotating two full cycles as this constant bias term for each transducer. The total wrench applied to the body by the legs under these conditions had 5N standard deviation, and the force was distributed from -5N to 20N, shown in figure 4.2 – nowhere near the expected gravitational wrench.

We collected F/T measurements and motion tracking data with BigANT standing at different poses. We positioned the robot in all $3^6 = 729$ combinations of three shaft angle possibilities for each leg: 0°(pointing straight down) or ±36°. We waited for 5 seconds between consecutive poses. When a pose transitioned to the next pose, we observed that both the wrench and motion

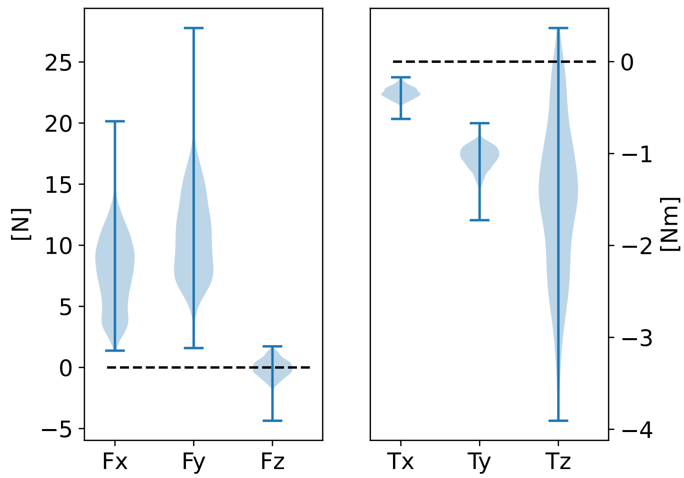

Figure 4.2: Violin plot of the sum of all wrenches acting on the robot when standing at different poses, with F/T transducers calibrated with naive offset removal.

tracking measurements exhibited an overshooting oscillatory behavior. We estimated the 1st order derivative of both data streams using a 2nd order `scipy.signal.savgol_filter` filter. We considered a pose to have reached steady state if the L2-norm of this numerical differentiation on force data decayed to less than 1 N/s. If a pose stayed at this "steady state" for more than $1.5s$, we took the median of the steady state measurements to represent this pose, thereby using the median filter's superior outlier rejection to guard against any non-zero mean noise processes. If a pose did not reach "steady state" we discarded that pose; this process left us with recordings from 635 poses. We further discarded those poses that did not have all 12 markers on the chassis tracked by motion capture system, leaving us with 543 poses. We transformed the transducer measurements from transducer frame into the floating base frame, where frame transformations were estimated by solving the orthogonal Procrustes problem using SVD on the 12 chassis markers.

The transducers themselves were pre-calibrated by ATI, and used in their normal operating region. The non-physical total body wrenches we obtained lead us to conclude that more a complete calibration procedure may be advisable for improving the accuracy of these measurements.

4.1.2.2 Background on F/T transducer calibration

ATI, the transducer manufacturer, instructs that we should expect an offset error for the transducers, i.e. that a nonzero reading should appear even when the transducer is unloaded and that this reading is added to all measurements made by that transducer. The ATI recommendation is for users to mechanically disconnect all transducers and re-calibrate them once a while. While this calibration could remove such an expected transducer offset,removing an reinstalling the transducers could introduce a different assembly error every time. Also, a fixed offset error cannot account for the gravity of a leg as it is moving through various positions. Hence, we decided to design an in-situ calibration procedure.

A common method to perform in-situ calibration of F/T transducers in robotic manipulators is to attach a test mass to the transducer, and then take several measurements with transducer at different orientations. Using the fact that the external force applied to transducer – the gravity of the attached mass – is fixed in the world frame, one can use least squares methods to estimate calibration parameters. The authors of [79, 67] solved for a in-situ calibration matrix which transformed strain gauge values to forces and torques. The authors of [80] solved for in-situ calibration matrix together with a rotation matrix between world frame and the robot base frame, to account for the error caused by a tilted robot base of a manipulator. The authors of [81] did gravity compensation for a tool together with estimating the transformation matrix between transducer frame and the robot frame. All of these researchers attached their F/T transducer rigidly fixed with respect to the end effector being loaded with the test mass.

4.1.2.3 Leg shape changes provide a challenge

In our robot, the CoM of each leg was not at a constant location in the body frame, but instead moved as a function of its drive shaft angle. We also observed that our placement of the transducers was not perfectly aligned and precise relative to the robot body frame, i.e. there were sometimes mounting position and orientation errors. Furthermore, the origin of the robot body frame was not the CoM of the robot body – only close to it. It is practically quite difficult to precisely estimate the CoM of a robot built and assembled with parts using many kinds of materials, and even with light legs this CoM will shift as the legs move.

The in-situ calibration procedures available in the robotic manipulation literature could assume near perfect positioning of the test mass by controlling the robot joints. The only reorientation available to us was to hang the robot from various attachment points to change the transducer frame orientations with respect to gravity. However, this hanging angle was itself difficult to measure in the body frame. Below, we present a calibration procedure that addresses all of the aforementioned challenges.

Besides these error sources, we also discovered that the transducer casing was not well isolated from forces applied to the exterior of this seemingly rigid metal casing. As an illustration of these issues, we wrapped 1-7 rubber bands (OfficeMax size 16) around the sensor case of an unloaded sensor. The transducer xy reading changed by as much as 1N. By gently pulling one rubber band to create a non-uniform load around the case, the readings from the transducer xy axis changed by 3N.

4.2 Force/torque sensing model

In this section, we present the notations, sensing error model and calibration parameters.

4.2.1 Notations

We use upper-case letters to represent matrices (e.g. T, R); and lower-case letters to represent vectors (e.g. w, r). We use the generalized force, wrench $w = [f; \tau]^T$, to represent linear components (forces $f \in \mathbb{R}^3$) and angular components (torques $\tau \in \mathbb{R}^3$). We use ; to separate elements in a column vector. We use $\mathsf{T}_\mathsf{A}^\mathsf{B}$ to represent rigid-body motion $\mathbf{SE(3)}$ from frame A to frame B. The homogeneous transformation representation is:

$$\mathsf{T}_\mathsf{A}^\mathsf{B} = \begin{bmatrix} \mathsf{R}_\mathsf{A}^\mathsf{B} & p_\mathsf{A}^\mathsf{B} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

Here $\mathsf{R}_\mathsf{A}^\mathsf{B} \in SO(3)$ is the rotation from frame A to B, and $p_\mathsf{A}^\mathsf{B} \in \mathbb{R}^3$ is the translation from the origin of A to the origin of B. With an abuse of notation, we omit 1 in the homogeneous transformation in the later calculations, i.e. $r^\mathsf{A} \in \mathbb{R}^3$, $\mathsf{T}_\mathsf{A}^\mathsf{B} r^\mathsf{A} := \mathsf{T}_\mathsf{A}^\mathsf{B}[r^\mathsf{A}; 1] = [r^\mathsf{B}; 1] =: r^\mathsf{B}$. We transform wrenches from frame A to frame B via the transposed adjoint representation of transformation $\mathsf{T}_\mathsf{B}^\mathsf{A}$:

$$\begin{aligned} w^\mathsf{B} &= [\mathsf{Ad}_{\mathsf{T}_\mathsf{B}^\mathsf{A}}]^T w^\mathsf{A} \\ \begin{bmatrix} f^\mathsf{B} \\ \tau^\mathsf{B} \end{bmatrix} &= \begin{bmatrix} (\mathsf{R}_\mathsf{B}^\mathsf{A})^T & \mathbf{0} \\ -(\mathsf{R}_\mathsf{B}^\mathsf{A})^T \left[p_\mathsf{B}^\mathsf{A}\right] & (\mathsf{R}_\mathsf{B}^\mathsf{A})^T \end{bmatrix} \begin{bmatrix} f^\mathsf{A} \\ \tau^\mathsf{A} \end{bmatrix} \end{aligned}$$

where the bracket $\left[p_\mathsf{A}^\mathsf{B}\right]$ lifts the 3d vector p_A^B to a skew symmetric matrix. The frame of reference for a value of interest is on its upper right corner. We use k to denote the index of the legs, i.e. wrench w.r.t. frame A on leg k is w_k^A.

The coordinate frames of interests are B: robot body frame; C: robot body frame with centered at CoM; W: floating base frame; ft_k: F/T transducer frame of the k^{th} leg. We use $g = [0, 0, \mathsf{g}]$ to represent the gravitational acceleration in the floating base frame.

We use ϕ to represent the shaft angle governing the shape of four-bar linkage leg, m to represent the mass of a leg, and $r(\phi)$ to represent the leg CoM as a function of ϕ. We use $i = 1, \cdots, N$ to denote index for time series. We use $\square^*$ to denote the optimal value for the variable in optimization. In §4.3.1, where we do the hanging experiment to estimate transducer and leg gravity offsets, we use $\square^\downarrow$ and $\square^\uparrow$ to denote a value associated with positive and negative hanging direction. We use $\square^- =: (\square^\downarrow - \square^\uparrow)/2$ and $\square^+ =: (\square^\downarrow + \square^\uparrow)/2$ to denote subtraction and addition of some value measured in two opposite hanging directions.

4.2.2 Sensing model

From our experiments, we noticed that for each transducer there existed a constant offset, independent of load, which was also mentioned in ATI F/T transducer manual [78]. We modeled the measured wrench $w_m^{\mathrm{ft_k}}$ by the $\mathrm{k^{th}}$ force/torque transducer, as the sum of the applied wrench $w_a^{\mathrm{ft_k}}$, an unknown transducer offset $w_o^{\mathrm{ft_k}} \in \mathbb{R}^6$ and random noise n,

$$w_m^{\mathrm{ft_k}} =: w_a^{\mathrm{ft_k}} + w_o^{\mathrm{ft_k}} + n \tag{4.1}$$

Since the transducers were installed underneath the legs, when a leg was in swinging phase (not contacting the ground), the wrench resulted from the leg gravity was applied to the transducer. We decomposed the applied wrench $w_a^{\mathrm{ft_k}}$ into two parts: the wrench resulted from the gravity of $\mathrm{k^{th}}$ leg and the ground contact wrench. The shaft angle ϕ_{k} governed the shape of four-bar linkage of each leg, and hence changed its CoM. We modeled the $\mathrm{k^{th}}$ leg CoM position in transducer frame by $r_{\mathrm{k}}^{\mathrm{ft_k}}(\phi_{\mathrm{k}})$. We used leg gravitational offset $w_{\mathrm{leg,k}}^{\mathsf{W}}(\phi_{\mathrm{k}})$, a function of ϕ_{k}, to model its gravity, and the torque generated by gravity. The gravitational acceleration is constant in the floating base frame. We transformed leg CoM position in the transducer frame to the floating base frame, and leg gravitational offset wrench:

$$w_{\mathrm{leg,k}}^{\mathsf{W}}(\phi_{\mathrm{k}}) =: [m_k g; \mathsf{T}_{\mathrm{ft_k}}^{\mathsf{W}} r_{\mathrm{k}}^{\mathrm{ft_k}}(\phi_{\mathrm{k}}) \times m_k g]$$

We decomposed the transformation from the $\mathrm{k^{th}}$ F/T transducer to floating base frame in four steps:

$$\mathsf{T}_{\mathrm{ft_k}}^{\mathsf{W}} =: \mathsf{T}_{\mathsf{C}}^{\mathsf{W}} \mathsf{T}_{\mathrm{ft_k}}^{\mathsf{C}} = \mathsf{T}_{\mathsf{C}}^{\mathsf{W}} \mathsf{T}_{\mathsf{B}}^{\mathsf{C}} \tilde{\mathsf{T}}_k \mathsf{T}_{\mathrm{ft_k}}^{\mathsf{B}} \tag{4.2}$$

Read from right to left, we had:

1. A nominal transformation, $\mathsf{T}_{\mathrm{ft_k}}^{\mathsf{B}}$, from the $\mathrm{k^{th}}$ transducer frame to body frame, calculated from robot design files;

2. Unknown transformation $\tilde{\mathsf{T}}_k$ to compensate the installation or manufacturing error, which was not captured by step 1;

3. Unknown translation $\mathsf{T}_{\mathsf{B}}^{\mathsf{C}}$ between the origin of the body frame and robot CoM.

4. A pure rotation $\mathsf{T}_{\mathsf{C}}^{\mathsf{W}}$ estimated from markers measured by motion tracking system, rotating the body frame centered at CoM into floating base frame.

Let $g^{\mathsf{C}} = \mathsf{R}_{\mathsf{W}}^{\mathsf{C}} g$ be the gravitational acceleration rotated into body CoM frame. We used $w_{\mathrm{gc},k}^{\mathsf{C}}(\phi_{\mathrm{k}})$ to denote the ground contact wrench in the body CoM frame, calculated from the applied wrench $w_a^{\mathrm{ft_k}}$ transformed in the CoM frame and subtracted by the wrench from leg gravity.

$$
\begin{aligned}
w^{\mathsf{C}}_{\mathrm{gc},k}(\phi_{\mathrm{k}}, g^{\mathsf{C}}) &=: [\mathsf{Ad}_{(\mathsf{T}^{\mathsf{C}}_{\mathrm{ft_k}})^{-1}}]^T w_a^{\mathrm{ft_k}} - w^{\mathsf{C}}_{\mathrm{leg,k}}(\phi_{\mathrm{k}}, g^{\mathsf{C}}) \\
\text{(substitute } \mathsf{T}^{\mathrm{ft_k}}_{\mathsf{C}} \text{ by eqn.4.2)} &= [\mathsf{Ad}_{(\mathsf{T}^{\mathsf{C}}_{\mathsf{B}}\tilde{\mathsf{T}}_k \mathsf{T}^{\mathsf{B}}_{\mathrm{ft_k}})^{-1}}]^T w_a^{\mathrm{ft_k}} - w^{\mathsf{C}}_{\mathrm{leg,k}}(\phi_{\mathrm{k}}, g^{\mathsf{C}}) \\
\text{(substitute } w_a^{\mathrm{ft_k}} \text{ by eqn.4.1)} &= [\mathsf{Ad}_{(\mathsf{T}^{\mathsf{C}}_{\mathsf{B}}\tilde{\mathsf{T}}_{\mathrm{k}} \mathsf{T}^{\mathsf{B}}_{\mathrm{ft_k}})^{-1}}]^T (w_m^{\mathrm{ft_k}} - w_o^{\mathrm{ft_k}}) \\
\text{(rewrite } w_o^{\mathrm{ft_k}} \text{ in F/T elements)} &\qquad - [m_k g^{\mathsf{C}}; \mathsf{T}^{\mathsf{C}}_{\mathsf{B}}\tilde{\mathsf{T}}_k \mathsf{T}^{\mathrm{k}}_{\mathrm{ft_k}}{}^{\mathrm{ft_k}}(\phi_{\mathrm{k}}) \times m_k g^{\mathsf{C}}]
\end{aligned}
\tag{4.3}
$$

It could be transformed into the floating base frame simply by the motion tracking rotation $[\mathsf{Ad}_{\mathsf{T}^{\mathsf{C}}_{\mathsf{W}}}]^T$. All the unknown calibration parameters were highlighted in eqn. (4.3). In the rest of the paper, leg CoM $r_{\mathrm{k}}(\phi_{\mathrm{k}})$ was always considered to be in the transducer $\mathrm{ft_k}$ frame, associated with the $\mathrm{k^{th}}$ shaft angle, so we omitted $\mathrm{ft_k}$ the leg CoM and k in ϕ_k, as $r_{\mathrm{k}}(\phi)$ to simplify the notation.

4.3 Calibration method

In this section, we documented our method to infer parameters by using several optimizations applied to measurement data. We formulated our parameter estimation goal function by using the fact that the force of gravity on the robot and its legs must be constant in the floating base frame. Our calibration had two steps: (1) we estimated the transducer offset $w_o^{\mathrm{ft_k}}$ and leg gravity offset $w^{\mathsf{W}}_{\mathrm{leg,k}}$, by summing up the measurements with leg gravity acting in opposite directions in the transducer frame. Because perfectly opposite measurements were never possible in reality, we assumed the existence of a small rotational term which could leak $m_k g$ into the other two directions, and bias the estimation of offsets. We modeled this small rotation error as the identity plus a small skew symmetric matrix, and solved for the 3D skew part together with other parameters. (2) we estimated the transformation error together with the unknown translation between body frame origin and CoM $\mathsf{T}^{\mathsf{C}}_{\mathsf{B}}\tilde{\mathsf{T}}_k$, by having the robot standing at different poses, then optimizing the error between the sum of all ground contact wrenches in floating frame and $[0, 0, G_{\mathrm{robot}}, 0, 0, 0]^T$, namely the robot gravity with zero total torque.

4.3.1 Transducer offset and leg gravity offset

To model the wrench applied by gravity to the a leg with respect to its transducer, we estimated the leg CoM as a function of shaft angle. Because this measurement is influenced by the transducer offset, we estimated the CoM function of shaft angle together with the transducer offset. We collected measurements with the robot hanging (zero ground contact wrenches) in opposite directions, while at the same time slowly (quasi-statically) rotating each leg ten full cycles with 9000 sample points for each cycle (see e.g. opposite z-directions in figure 4.3). The torque that resulted from leg gravity arises from a cross-product with leg CoM, leaving a one-dimension null space when estimating from each pair of hanging experiments (more details in §4.3.1.3). To get a complete 3D CoM function of shaft angle, we repeated the procedure twice: two pairs of opposite orientations, one along the transducer $\pm x$-axis, and one along the transducer $\pm z$-axis (aligning with gravity).

In the rest of this section, we used the experiments of transducer $\pm x$-axis aligning with gravity to explain the calculation; the calculation with z-axis positive or negative direction aligning with gravity could be performed analogously.

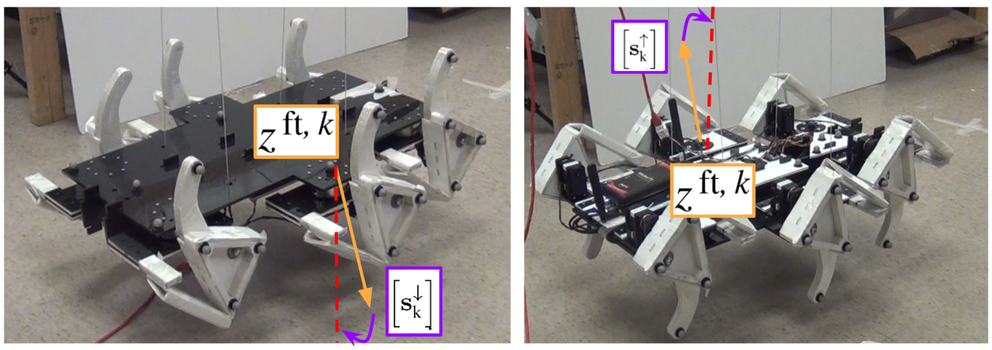

Figure 4.3: BigAnt robot hanging with transducer z-aixs positive $z^{\mathrm{ft_k}}$ (orange) and negative direction aligning with gravity direction (red); $\left[s_{\mathrm{k}}^{\downarrow}\right]$ and $\left[s_{\mathrm{k}}^{\uparrow}\right]$ (purple): skew symmetric matrices modeling the hanging orientation error.

4.3.1.1 Hanging measurement model

We considered a measurement model using a skew symmetric matrix $[s_{\mathrm{k}}]$, lifted from $s_{\mathrm{k}} \in \mathbb{R}^3$, to model the axes misalignment error between the direction of gravity and the transducer axis. We assumed s_{k} was constant throughout one time series, i.e. the transducer orientation did not change throughout the measurement. The skew symmetric matrix $[s_{\mathrm{k}}]$, the tangent space at identity rotation, could be viewed as an infinitesimally small rotation. The applied wrench, when leg gravity aligning with positive x-axis of the transducer frame, became

$$f_a^{\mathrm{ft_k},\downarrow} = (\left[s_{\mathrm{k}}^{\downarrow}\right] + \mathrm{I})m_{\mathrm{k}}g_x; \;\; \tau_a^{\mathrm{ft_k},\downarrow} = [r_{\mathrm{k}}(\phi)]\,(\left[s_{\mathrm{k}}^{\downarrow}\right] + \mathrm{I})m_{\mathrm{k}}g_x$$

Its opposite applied wrench, when leg gravity aligning with negative x-axis of the transducer frame, was the same magnitude with opposite direction.

$$f_a^{\mathrm{ft_k},\uparrow} = -(\left[s_{\mathrm{k}}^{\uparrow}\right] + \mathrm{I})m_{k}g_x; \;\; \tau_a^{\mathrm{ft_k},\uparrow} = -\,[r_{\mathrm{k}}(\phi)]\,(\left[s_{\mathrm{k}}^{\uparrow}\right] + \mathrm{I})m_{\mathrm{k}}g_x$$

Here $\left[s_{\mathrm{k}}^{\downarrow}\right]$ and $\left[s_{\mathrm{k}}^{\uparrow}\right]$ were two skew symmetric matrices, accounting for two different small rotations between F/T transducer positive or negative x-axis and world z-axis. During the experiment, we measured F/T data for both of these two configurations, with the same set of varying shaft angles ϕ_i, for $i = 1, \cdots, N$, covering the full range of motion multiple times. We modeled the measured wrenches, according to eqn. (4.1), and partitioned them into their force and torque components in eqn. (4.4)-4.7, where $\nu_i^{\downarrow}, \nu_i^{\uparrow}, \omega_i^{\downarrow}, \omega_i^{\uparrow}$ were realization of random noise for the i^{th} measurement.

$$f_{m,i}^{\mathrm{ft_k},\downarrow} = (\left[s_{\mathrm{k}}^{\downarrow}\right] + \mathrm{I})m_{\mathrm{k}}g_x + f_o^{\mathrm{ft_k}} + \nu_i^{\downarrow} \tag{4.4}$$

$$f_{m,i}^{\mathrm{ft_k},\uparrow} = -(\left[s_{\mathrm{k}}^{\uparrow}\right] + \mathrm{I})m_k g_x + f_o^{\mathrm{ft_k}} + \nu_i^{\uparrow} \tag{4.5}$$

$$\tau_{m,i}^{\mathrm{ft_k},\downarrow} = [r_{\mathrm{k}}(\phi_i)]\,(\left[s_{\mathrm{k}}^{\downarrow}\right] + \mathrm{I})m_{\mathrm{k}}g_x + \tau_o^{\mathrm{ft_k}} + \omega_i^{\downarrow} \tag{4.6}$$

$$\tau_{m,i}^{\mathrm{ft_k},\uparrow} = -\,[r_{\mathrm{k}}(\phi_i)]\,(\left[s_{\mathrm{k}}^{\uparrow}\right] + \mathrm{I})m_{\mathrm{k}}g_x + \tau_o^{\mathrm{ft_k}} + \omega_i^{\uparrow} \tag{4.7}$$

4.3.1.2 Leg gravity, $m_k g$

We subtracted two opposite force measurements between eqn. (4.4) and eqn. (4.5). By dividing the difference by 2, we got the leg gravity multiplied by a small rotation error term in the front.

$$\frac{(4.4)-(4.5)}{2}: \quad f_{m,i}^{\mathrm{ft_k},-} = \left(\frac{\left[s_{\mathrm{k}}^{\downarrow}\right] + \left[s_{\mathrm{k}}^{\uparrow}\right]}{2} + \mathrm{I}\right)m_{\mathrm{k}}g_x + \frac{\nu_i^{\downarrow} - \nu_i^{\uparrow}}{2}$$

The sum of two skew symmetric matrices is still skew symmetric, denoted by $\left[s_{\mathrm{k}}^{+}\right] = (\left[s_k^{\downarrow}\right] + \left[s_k^{\uparrow}\right])/2$. Ideally, $m_{\mathrm{k}}g_x$ was equal to $[m_{\mathrm{k}}\mathsf{g}, 0, 0]$, a constant value in x component, and zeros in y, z components. Taking that as the optimization goal, with N measurement samples $f_{m,i}^{\mathrm{ft_k},-}$ for $i = 1, \ldots, N$, we solved for s_{k}^{+} using `scipy.optimize.least_squares` with the Trust Region Reflective algorithm and cost function tolerance 10^{-3}.

$$\begin{aligned} s_{\mathrm{k}}^{+,*} = \arg\min_{s_{\mathrm{k}}^{+}} \frac{1}{\sqrt{N}} \sum_{j=2,3} \sqrt{\sum_{i=1}^{N}((\left[s_{\mathrm{k}}^{+}\right] + \mathrm{I})^{-1} f_{m,i}^{\mathrm{ft_k},-} \cdot \mathbf{e}_j)^2} \\ + \operatorname{std}_{i=1,\cdots,N}((\left[s_{\mathrm{k}}^{+}\right] + \mathrm{I})^{-1} f_{m,i}^{\mathrm{ft_k},-} \cdot \mathbf{e}_1) \end{aligned}$$

where, $\mathbf{e}_j$ with $j = 1, 2, 3$ are unit vectors in $\mathbb{R}^3$. We estimated the gravity of the leg by

$$m_k g = 1/N \sum_{i=1}^{N}((\left[s_{\mathrm{k}}^{+,*}\right] + \mathrm{I})^{-1} \tilde{f}_{m,i}^{\mathrm{ft_k}} \cdot \mathbf{e}_1)$$

4.3.1.3 Leg CoM, $r_\mathrm{k}(\phi)$

Next, we solved for the leg CoM by subtracting eqn. (4.6) and eqn. (4.7).

$$\begin{aligned}\frac{(4.6)-(4.7)}{2}: \quad \tau_{m,i}^{\mathrm{ft_k},-} &= [r_\mathrm{k}(\phi_i)]\left(\left[s_\mathrm{k}^{+}\right]+\mathrm{I}\right)m_\mathrm{k}g_x+\omega_i^{-} \\ &= [r_\mathrm{k}(\phi_i)]\, f_{m,i}^{\mathrm{ft_k},-}+\omega_i^{-}\end{aligned}$$

Here $\tau_{m,i}^{\mathrm{ft_k},-}$ and $f_{m,i}^{\mathrm{ft_k},-}$ could be calculated directly from the measurement data. We estimated the leg CoM $r_{\mathrm{est,k}}(\phi)$ as a function of ϕ through a Kalman smoother [82], with N samples measured with varying shaft angle ϕ_i, $i=1,\cdots,N$. We set the transition and observation covariance matrices in the Kalman smoother both having 0.01 on the diagonal and 0 on the off diagonal elements, which matched the magnitude of the measured covariance. Since the torque was the cross product between CoM vector and gravity, the estimated CoM here had a one dimensional null space along the direction of $\tilde{f}_{m,i}^{\mathrm{ft_k}}$. The CoM could be $r_\mathrm{k}(\phi_i)=r_{\mathrm{est,k}}(\phi_i)+p_i\cdot\tilde{f}_{m,i}^{\mathrm{ft_k}}$, for any scalar $p_i\in\mathbb{R}$. Therefore, we needed at least another pair of opposite hanging measurements to get full rank information on $r_\mathrm{k}(\phi)$. In our experiment, we took measurements with leg gravity aligning with $\pm z$-axis of the transducer frame and performed analogous calculations mentioned above with g_z. We obtained another estimation $r'_{\mathrm{est,k}}(\phi)$ where $r_\mathrm{k}(\phi_i)=r'_{\mathrm{est,k}}(\phi_i)+p'_i\cdot\tilde{f}'^{\mathrm{ft_k}}_{m,i}$. We solved p_i, p'_i by equating two $r_\mathrm{k}(\phi)$ obtained from two sets of measurements. It formed an over-determined system with three equations and two unknowns, so we solved for the unknowns p_i, p'_i by ordinary least squares. We averaged those two r_k and fit a function with respect to ϕ.

4.3.1.4 Transducer offset, $w_o^{\mathrm{ft_k}}=[f_o^{\mathrm{ft_k}};\tau_o^{\mathrm{ft_k}}]$

Then, we solved for transducer torque offset $\tau_o^{\mathrm{ft_k}}$, by adding eqn. (4.6) and eqn. (4.7).

$$\begin{aligned}\frac{(4.6)+(4.7)}{2}: \quad \tau_{m,i}^{\mathrm{ft_k},+} &= [r_\mathrm{k}(\phi_i)]\left(\frac{\left[s_\mathrm{k}^{\downarrow}\right]-\left[s_\mathrm{k}^{\uparrow}\right]}{2}\right)m_k g_x \\ &\quad +\tau_o^{\mathrm{ft_k}}+\frac{\omega_i^{\downarrow}+\omega_i^{\uparrow}}{2}\end{aligned}$$

We denoted the subtraction between two skew symmetric matrices by $\left[s_\mathrm{k}^{-}\right]=\left(\left[s_\mathrm{k}^{\downarrow}\right]-\left[s_\mathrm{k}^{\uparrow}\right]\right)/2$. In the transducer model in eqn. (4.1), we assumed the transducer offset was constant. We solved for the optimal value of s_k^{-} by minimizing the standard deviation of $\tau_o^{\mathrm{ft_k}}$ over all samples:

$$s_\mathrm{k}^{-,*}=\arg\min_{s_\mathrm{k}^{-}}\ \underset{i=1,\cdots,N}{\mathrm{std}}\left(\tau_{m,i}^{\mathrm{ft_k},+}-[r_\mathrm{k}(\phi_i)]\left[s_\mathrm{k}^{-}\right]m_\mathrm{k}g_x\right)$$

Finally, we calculated the force and torque offsets by:

$$f_o^{\mathrm{ft_k}} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f_{m,i}^{\mathrm{ft_k},\downarrow} + f_{m,i}^{\mathrm{ft_k},\uparrow}}{2} - \left[s_{\mathrm{k}}^{-,*}\right] m_{\mathrm{k}} g_x\right)$$

$$\tau_o^{\mathrm{ft_k}} = \frac{1}{N}\sum_{i=1}^{N}\left(\tau_{m,i}^{\mathrm{ft_k},+} - [r_{\mathrm{k}}(\phi_i)]\left[s_{\mathrm{k}}^{-,*}\right] m_{\mathrm{k}} g_x\right)$$

4.3.2 Transformation error, $\mathsf{T}_{\mathsf{B}}^{\mathsf{C}}\tilde{\mathsf{T}}_k$

In the second part, we showed our method in estimating the transformation error from transducer frame to body frame, $\tilde{T}_{\mathrm{k}}$ modeled by $(\tilde{\mathsf{R}}_{\mathrm{k}}, \tilde{p}_{\mathrm{k}})$, together with an unknown translation between body frame and CoM frame $\mathsf{T}_{\mathsf{B}}^{\mathsf{C}}$ modeled by $(\mathrm{I}, p_{\mathsf{B}}^{\mathsf{C}})$. We used the fact that the robot should be at force-torque balance when standing still at different poses. At each pose, the sum of the ground contact wrenches from all feet, transformed to the floating base frame, should equal to the gravity of the robot with zero torque, as calculated below:

$$\begin{aligned}
w_{\mathrm{total},p}^{\mathsf{W}} &= \sum_{k=1}^{K}[\mathsf{Ad}_{(\mathsf{T}_{\mathsf{C},p}^{\mathsf{W}})^{-1}}]^T w_{\mathrm{gc},k}^{\mathsf{C}}(\phi_{p,\mathrm{k}}, g_p^{\mathsf{C}}) \\
&= \sum_{k=1}^{K}[\mathsf{Ad}_{(\mathsf{T}_{\mathsf{C},p}^{\mathsf{W}})^{-1}}]^T \Big(\mathsf{Ad}_{(\mathsf{T}_{\mathsf{B}}^{\mathsf{C}}\tilde{\mathsf{T}}_k \mathsf{T}_{\mathrm{ft_k}}^{\mathsf{B}})^{-1}}]^T (w_{m,p}^{\mathrm{ft_k}} - w_o^{\mathrm{ft_k}}) \\
&\qquad -[m_k g_p^{\mathsf{C}}; \mathsf{T}_{\mathsf{B}}^{\mathsf{C}}\tilde{\mathsf{T}}_k \mathsf{T}_{\mathrm{ft_k}}^{\mathsf{B}} \mathbf{r}_{\mathrm{k}}^{\mathrm{ft_k}}(\phi_{p,\mathrm{k}}) \times m_k g_p^{\mathsf{C}}]\Big) \qquad \text{(substitute } w_{\mathrm{gc},k}^{\mathsf{C}} \text{ by eqn.4.3)} \\
&= \sum_{k=1}^{K}\Bigg(\begin{bmatrix} \mathsf{R}_{\mathsf{C}}^{\mathsf{W}} & \mathbf{0} \\ \mathbf{0} & \mathsf{R}_{\mathsf{C}}^{\mathsf{W}} \end{bmatrix}\begin{bmatrix} \tilde{\mathsf{R}}_k & \mathbf{0} \\ \tilde{\mathsf{R}}_{\mathrm{k}}\left[\tilde{\mathsf{R}}_k(\tilde{p}_{\mathrm{k}} + p_{\mathsf{B}}^{\mathsf{C}})\right] & \tilde{\mathsf{R}}_{\mathrm{k}} \end{bmatrix}[\mathsf{Ad}_{(\mathsf{T}_{\mathrm{ft_k}}^{\mathsf{B}})^{-1}}]^T (w_{m,p}^{\mathrm{ft_k}} - w_o^{\mathrm{ft_k}}) \\
&\qquad -[m_k g; \begin{bmatrix} \mathsf{R}_{\mathsf{C}}^{\mathsf{W}} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}\begin{bmatrix} \tilde{\mathsf{R}}_k & \tilde{p}_{\mathrm{k}} + p_{\mathsf{B}}^{\mathsf{C}} \\ \mathbf{0} & 1 \end{bmatrix}\mathsf{T}_{\mathrm{ft_k}}^{\mathsf{B}} \mathbf{r}_{\mathrm{k}}^{\mathrm{ft_k}}(\phi_{p,\mathrm{k}}) \times m_k g]\Bigg) \qquad \text{(Expand } \mathsf{T}_{\mathsf{B}}^{\mathsf{C}}\tilde{T}_k\text{)} \\
&\approx [0, 0, G_{\mathrm{robot}}, 0, 0, 0]^T
\end{aligned}$$

We used exactly the same dataset as described in §4.1.2.1, and inferred the unknown parameters by minimizing the error between the total ground contact wrenches and the gravity of the robot with zero torque, as shown in eqn. (4.8). It minimized the difference between the measured and expected total wrench on CoM over all $p = 1, \cdots, P$ poses. The K-norm we used had a positive definite diagonal matrix K to allow us to compare loss functions computed over force and torque, which had different physical units. In practice, we used 1, 1, 1, 10, 10, 10 in the diagonal elements and zeros elsewhere as the K-norm, so the forces and torques had about the same magnitude.

$$\min_{\tilde{\mathsf{R}}_k, \tilde{p}_k, p^{\mathsf{C}}_{\mathsf{B}}} \sum_{p=1}^{P} ||(w^{\mathsf{W}}_{\text{total},p} - [0, 0, G_{\text{robot}}, 0, 0, 0]^T)||^2_{\mathsf{K}} \tag{4.8}$$

$$\text{s.t. } \tilde{\mathsf{R}}_k \in SO(3)$$

As formulated, the optimization problem needed to solve for the six rotation matrices $\tilde{\mathsf{R}}_{\in SO(3)}$ the 3D rotation group, which is not a vector space. However, most numerical optimization paradigms only work on vector spaces. Instead, we parameterized the rotation using Cayley transformation: $\tilde{\mathsf{R}}_k = (\mathrm{I} - [\tilde{q}_k])(\mathrm{I} + [\tilde{q}_k])^{-1}$ [83]. This can represent rotation matrices over a wide range using a parameter space which is a vector space.

From here we made a change of variables $\tilde{p}'_k = \tilde{\mathsf{R}}_k(\tilde{p}_k + p^{\mathsf{C}}_{\mathsf{B}})$. This made the objective function of the optimization eqn. (4.8) into a bilinear form in $\tilde{p}'_k$ and $\tilde{\mathsf{R}}_k$. To solve this optimization we used dual least squares optimization, fixing one unknown and solving for the other, alternating which variable is fixed at each step, until it convergence. For each of these steps we again used the `scipy.optimize.least_squares` Trust Region Reflective algorithm with cost function tolerance 10^{-3}. Our dual optimization convergence criterion was testing when the maximum norm difference between two successive steps was less than 10^{-3}.

4.4 Calibration results

In this section, we reported the estimated calibration parameters and the residual after calibration. We used violin plots to show the distribution of the residuals, and all violin plots in this chapter used 100 points for the Gaussian kernel density estimation.

4.4.1 Transducer offset and leg gravity offset

The estimated transducer offset and the leg gravity were summarized in table 4.1. We plotted the estimated leg CoM trajectories of all legs in figure 4.4. After removing the transducer offsets and the leg gravity offsets, we plotted of the residual of all four orientations in figure 4.5. We compared our formulation with the naive addition and subtraction, without the axis misalignment error $\hat{s}$.

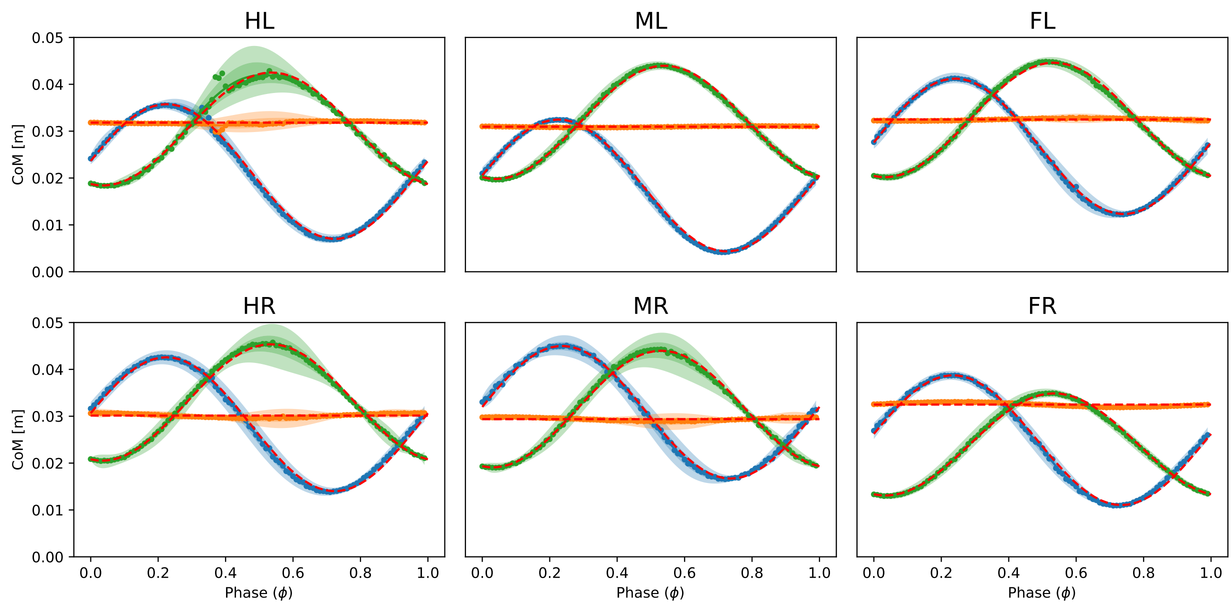

Figure 4.4: Estimated center of mass (CoM) vs. phase of all six legs. We plotted the median estimated CoM of ten cycles in dots, interquartile range in dark shaded region and 95% confidence interval in shallow shaded region (x coordinate in blue, y coordinate in orange and z coordinate in green). We fitted x, z coordinate with function $r(\phi) = A\sin(2\pi\phi + C) + D$, y coordinate using a constant, and plotted them in red dashes.

Table 4.1: Summary of the estimated force/torque offsets and the estimated gravity of legs. The top row is the abbreviation of the transducer underneath which leg. (FL: front left, ML: mid left, HL: hind left, FR: front right, MR: mid right, HR: hind right)

	HL	ML	FL	HR	MR	FR
$f_{\text{offset}}[N]$	1.47 ± 0.03	1.419 ± 0.001	0.57 ± 0.01	-8.49 ± 0.02	-0.06 ± 0.02	-0.93 ± 0.01
	-1.30 ± 0.01	-4.884 ± 0.001	3.71 ± 0.01	-1.448 ± 0.004	-0.123 ± 0.001	0.653 ± 0.001
	-5.61 ± 0.01	-0.406 ± 0.001	-2.039 ± 0.003	-2.43 ± 0.01	-0.07 ± 0.01	-2.183 ± 0.004
$\tau_{\text{offset}}[Nm]$	0.2457 ± 0.0002	-0.0136 ± 0.0002	0.2176 ± 0.0001	-0.3159 ± 0.0002	-0.2392 ± 0.0009	-0.3611 ± 0.0004
	0.002 ± 0.003	0.0808 ± 0.0002	0.0921 ± 0.0005	0.3449 ± 0.0009	-0.0982 ± 0.0012	-0.153 ± 0.001
	0.034 ± 0.003	0.0144 ± 0.0001	-0.0177 ± 0.0004	0.001 ± 0.001	0.004 ± 0.001	0.0222 ± 0.0007
$mg[N]$	6.2 ± 0.2	6.2 ± 0.2	6.2 ± 0.1	6.1 ± 0.2	6.2 ± 0.2	6.4 ± 0.1

4.4.2 Statistical analysis on wrench transformation error

We performed a model selection on 4 different models, to see the significance of the parameters used to model the wrench transformation error. We tested on

1. The full model, fitting all six rotations ($\tilde{\mathsf{R}}_k$) and six translations ($\tilde{p}'_k$), using $6 \times (3 + 3) = 36$ parameters;

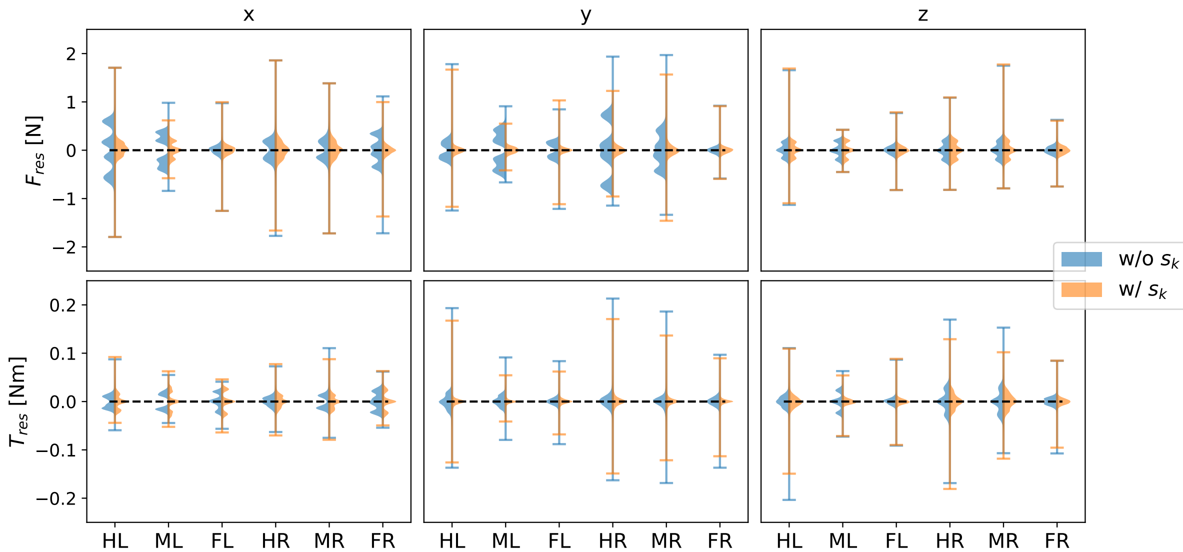

Figure 4.5: Violin plot of force (first row) and torque (second row) residual on xyz-axis (columns) for all six transducers. With skew symmetric matrices $\hat{s}$ accounting for hanging orientation error (orange), without (blue). (FL: front left, ML: mid left, HL: hind left, FR: front right, MR: mid right, HR: hind right).

2. No transducer translation error ($\tilde{p}_k = \mathbf{0}$) , only fitting six rotations ($\tilde{\mathsf{R}}_k$) and one single unknown translation ($p_{\mathsf{B}}^{\mathsf{C}}$) between robot center and CoM, using $6 \times 3 + 3 = 21$ parameters;

3. No transducer rotation error ($\tilde{\mathsf{R}}_k = \mathbf{I}_3$), only fitting the transducer translation error together with robot center to CoM translation ($\tilde{p}'_k$), using $6 \times 3 = 18$ parameters;

4. With neither transducer rotation, nor translation error, only fitting a single unknown translation between robot center and CoM, using 3 parameters.

The statistics of those models were summarized in table 4.2. We reported mean of the testing mean squared error (MSE) of 5-fold cross validation, together with Bayesian information criterion

Table 4.2: Summary of model selection statistics. (Abbreviations: degree of freedom(DOF), cross-validation(CV), Bayesian information criterion(BIC)

Model	DOF	CV error	BIC
Full (Six rotations + six translations)	18+18	12	1580
Six rotations + single translation	18+3	17	1674
Six translations only	18	62	2352
Single translation	3	105	2546

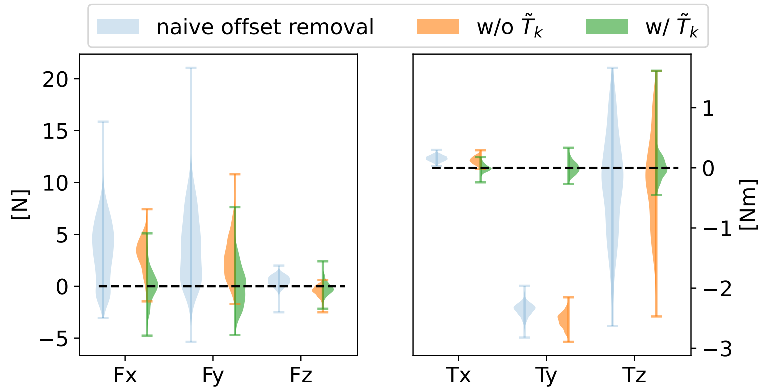

Figure 4.6: Violin plot of the residual of the sum of force (left) and torque (right) at robot CoM in floating base frame at different poses. Naive bias removal (light blue), without characterizing transformation error (orange), our calibration method with $(\tilde{\mathsf{R}}_k, \tilde{p}'_k)$ (green). Our calibration reduced the RSME (from blue to green) by 63% for forces (66% in Fx, 61% in Fy, 13% in Fz) and 90% for torques (61% in Tx, 95% in Ty, 78% in Tz).

$\mathrm{BIC} = P \ln(\mathrm{RSS}/P) + \mathrm{DOF} \times \ln P$, where RSS is the residual sum squared. We selected (1) full model using six rotations and six translations, which yielded the smallest BIC. We showed the residual plot comparing with and without transformation errors in figure 4.6.

Since the bilinear least-squares optimization only yielded local optima, we started the optimization process from 20 different randomized initial guesses within a plausible set, where the orientation error was less than 15 degrees in all three directions and the translation error was less one-third of the robot size. The cost function variation was only 3.2% among all initial conditions.

4.4.3 Tripod walking

We also collected the wrench and motion tracking data when the BigAnt robot used tripod gait to walk quasi-statically. The gait was governed by a Buehler clock running at frequency 0.08Hz and turned right at 9.08^o per gait cycle on average. A more detailed description of the gait can be found in [76, 8]. The measurements compared between naive bias taring and our calibration was shown below.

4.5 Discussions

In this paper, we showed that naive offset correction of the measurements from the six 6-axis transducers on our hexapedal robot in a static pose produced horizontal force estimates on the order of 25% of mg, and pitch and roll torque ranges that do not even contain the true torque of 0

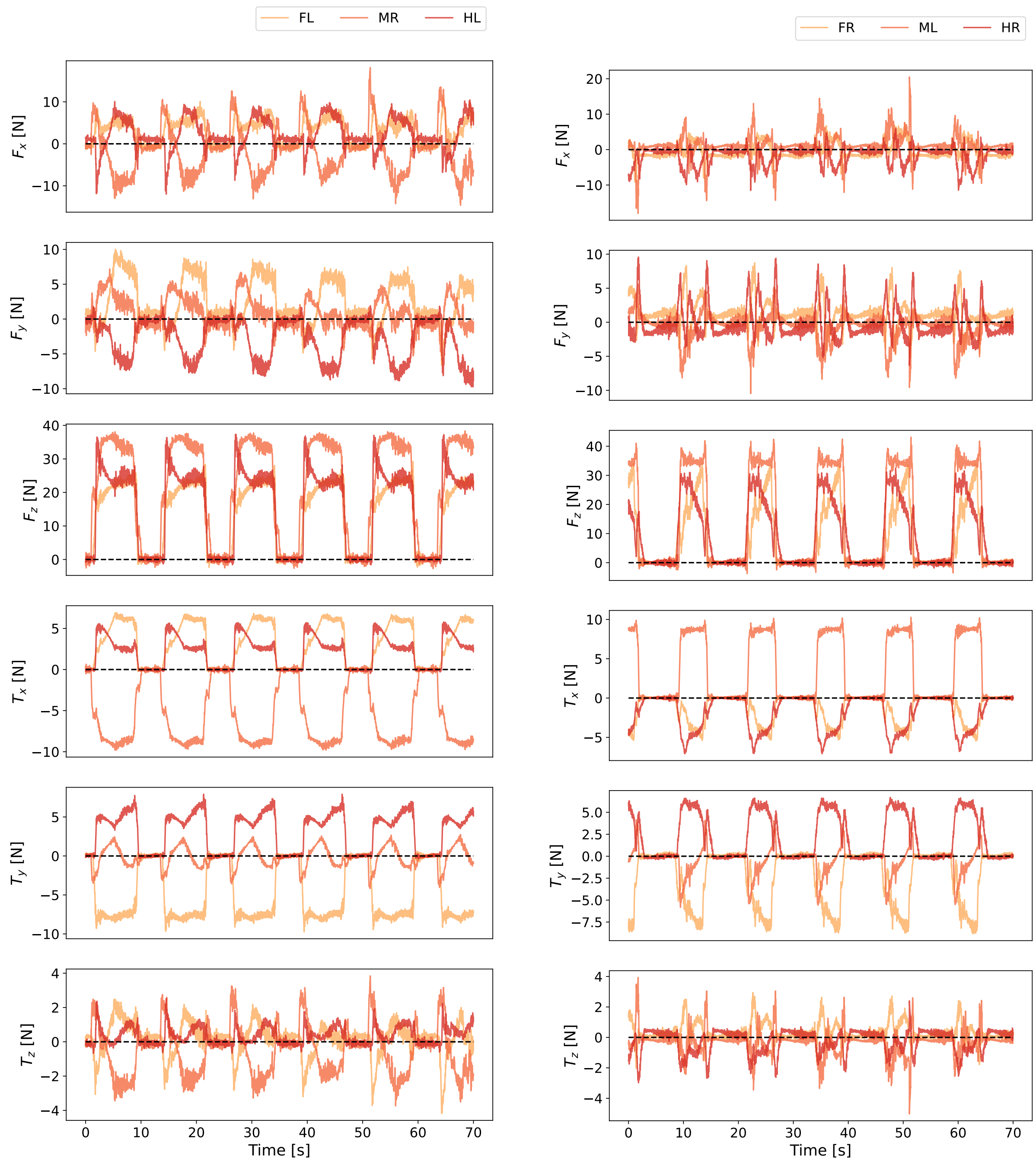

Figure 4.7: We plotted all groud contact wrenches on left tripod (left column) and right tripod (right column) measured from the f/t transducers with naive taring transformed to the floating base frame. (FL: front left, ML: mid left, HL: hind left, FR: front right, MR: mid right, HR: hind right)

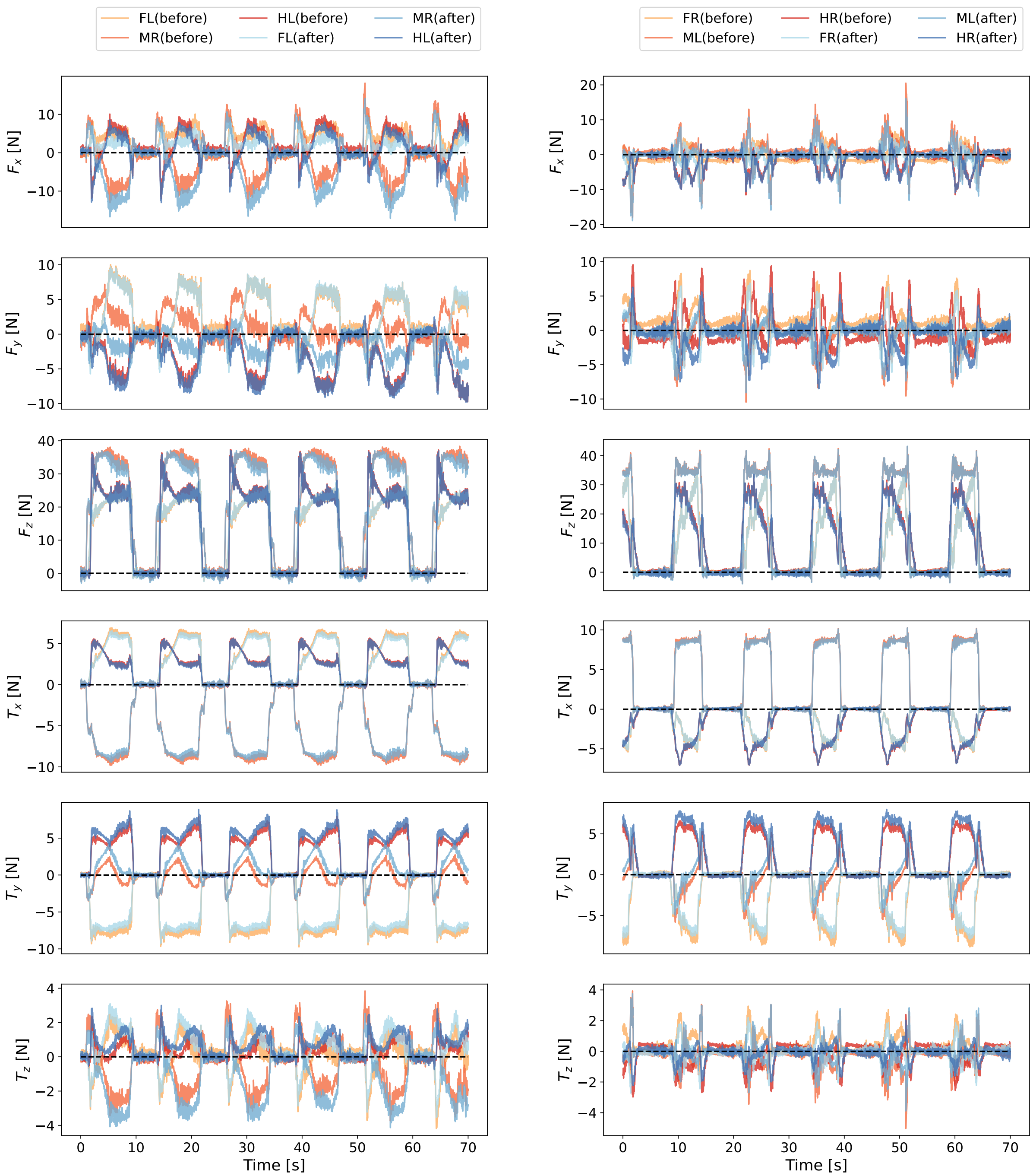

Figure 4.8: Right tripod.

Figure 4.9: We plotted all groud contact wrenches on left tripod (left column) and right tripod (right column) at the floating base frame after calibration compared with with naive taring.

(see figure 4.2). The poor accuracy of this ground contact wrench measurement hindered us from studying our model of multi-legged walking and slipping.

To resolve this we showed an in-situ calibration framework for these sensors which can be used without disassembling the robot. We characterized transducer offset, leg gravity offset and the wrench transformation error in our model.

We first estimated transducer offset and leg gravity offset by hanging the robot in two sets of opposite directions. Our calibration method took into account of a hanging orientation error $\hat{s}$ when solving for the transducer offset and tool gravitational offset. Without this $\hat{s}$ the residual error had a clearly multi-modal distribution (see figure 4.5), especially in Fx and Fy. By characterizing an infinitesimal rotation error $\hat{s}$ during the hanging experiment, the error distributions became more similar to a zero mean normal distribution, with a maximum 0.2N standard deviation.

In our second calibration step we characterized the frame transformation estimation error between transducer frame and body frame together with the unknown translation between body frame origin and CoM. The unknown translation between body frame origin and CoM removed the 2.5 Nm bias in torque estimation. Using the frame transformation correction, we reduced the 5-fold cross validation residual by a factor of ×8 compared with only fitting a single translation transformed to CoM. Overall, the calibration framework enabled us to make a much more accurate measurement of ground contact forces and the ensuing leg wrenches.

We also noticed that the offset term on the transducer could change. Our experience showed the offset changed up to 10 N after six months. Temperature change and gauge excitation voltage could be possible causes for the drift, should be expected over extended periods [78]. When such a significant offset change occurred, we easily re-estimated $w_o^{\mathrm{ft_k}}$ by putting it as the optimization variables in eqn. (4.8), with the other calibration parameters held unchanged. This only required a few measurements with robot standing at different poses. We found no need to redo the entire protocol of hanging experiments to adjust this calibration.

This calibration procedure can be generalized to other systems that have a tool attached to an F/T transducer. For example, when an F/T transducer is attached between the end of a manipulator and an actuated tool head, one can estimate the transducer offset and tool gravity offset by following the same procedure described in §4.3.1. However, instead of hanging, one should set the manipulator joint angles to position the tool in opposite orientations. The $\hat{s}$ term can compensate for imperfect opposite orientations, and the method used to characterize the leg CoM can be applied to determine the tool CoM during actuation.

The Cayley transformation parametrization used in §4.3.2 is a powerful technique to switch the optimization argument set from a rotation group to a vector space. It can be used to characterize unknown rotation matrices in general, as long as one can formulate an objective function to optimize towards.

4.6 Conclusion

We have demonstrated a calibration methodology that enabled us to simultaneously measure the ground contact wrenches in a six legged robot while it is walking and its legs are slipping. We used this calibration method to obtain the F/T measurements for the previous chapter, which discusses the multi-legged locomotion model with its predictions, and suggests some explanations as to why a non-Coulomb friction model is successful at predicting motion governed by Coulomb friction.

CHAPTER 5

Multi-legged Locomotion Model Extensions and Applications

5.1 Introduction

In chapter 3, we present an algorithm for multi-legged locomotion contacting forces and the body velocity together with slipping. The multi-legged systems are governed by principally kinematic motion, and we employ a friction law ansatz that allows us to compute the shape-change to body-velocity linear connection and the foot contact forces. The model is fast-to-compute, scales well with large number of contacts, each potentially with slipping, and it has been experimentally validated on several robots with different morphology. It enables the rapid and accurate estimation of multi-contact robot-environment interactions.

In this chapter, we seek to further improve and apply this model to real-world applications. In §5.2, we relax the model model assumption of flat surface, and allow the robot pitch and roll to vary around non-zero constant angles. This allows the model to apply to slopes and staircases. In §5.3, we use a data-driven connection residual term to capture the unmodeled dynamics, and improve the motion prediction on side velocities. We show this residual term is able to predict motion across gaits and frequencies. In §5.4, we implement this model on GPU, and plan the robot gait with a large number of possible motions computed in parallel. We show the paralleized computation allows us to exhaustively search among the entire configuration space, and generates multi-legged robot motions that are non-intuitive for human to design.

5.2 Uneven terrain

Natural terrain is rarely flat, and even some indoor area is also non-flat, for example staircases or ramps. Robots need to have the capability of traversing uneven terrains and stairs autonomously in urban disaster relief or search and rescue missions. Previous works have demonstrated that multi-legged robots with 1-DoF legs, Rhex and BigAnt, are robust to uneven terrain and able to climb up or down stairs with open loop control [84, 85, 86]. However, these gaits were usually tuned by engineers through experiments with respect to certain stair specification. Previous work on quadrupeds developed robust stair climbing algorithms through proprioceptive sensing [87] and trajectory optimization [88].

In the previous chapter §3.3, we assume that the ground is flat, and the robot body has small pitch and roll angles relative to 0. This assumption limits the usage of our proposed model on uneven surfaces, where many challenges arise for robots to navigates while maintaining efficient motion and stability. A simple, accurate and fast-to-compute model could help analyze and generate more reliable motion for robots to move through uneven terrain, and perhaps find the motion that minimizes slipping off the stairs. In this section, we generalize our model to accommodate to simple cases of uneven terrain: ramps and staircases. Specifically, instead of assuming small pitch and roll, we assume the pitch and roll are small with respect to some non-zero constant initial pitch (β_0) and roll (α_0); and instead of assuming the terrain is flat, we consider two cases: (1) the terrain has a constant slope, (2) terrain can be modeled as a piece-wise constant step function.

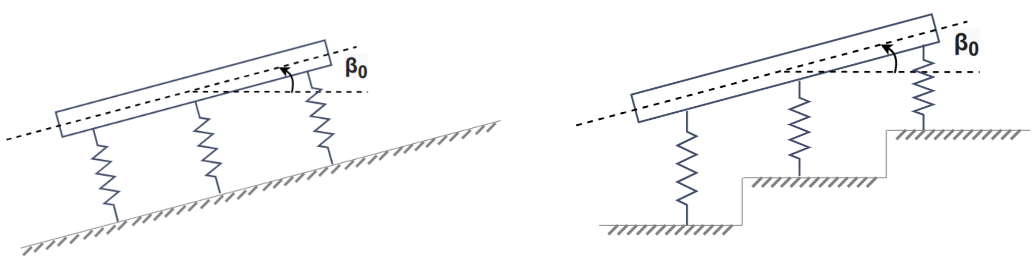

Figure 5.1: Schematic drawing of the 2D support spring model extended to slope (left) and stair cases (right). The robot is assumed to have a small pitch around the initial pitching value β_0.

5.2.1 Modeling the motion on a constant slope

When the surface has a constant known slope, assume rotation $R_0 \in SO(3)$ transform the horizontal plane (with respect to gravity) to the slope. In this case, we decompose the gravity into the directions that parallel and perpendicular to the slope. The total force balance becomes

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = R_0 \begin{bmatrix} 0 \\ 0 \\ Mg \end{bmatrix}$$

i.e. the spring-supported model searches for the solution satisfying force balance that vertical to the slope; and the sum of planar friction forces needs to compensate for the gravity instead of 0.

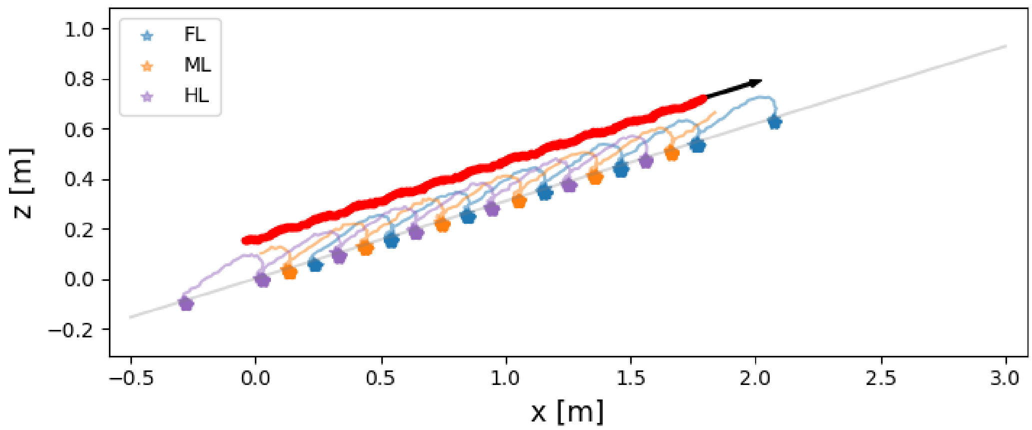

Figure 5.2: A trajectory of simulated BigAnt robot moving up 17^o slope. We plotted the slope (gray), robot center (red), the trajectories of front-left (FL, blue), mid-left (ML, orange), hid-left (HL, purple) feet before compress (stance: stars, swing: lines).

5.2.2 Modeling the motion on stairs

When moving to the stairs, we assume the robot pitch (β) has a small variation around the slope the stairs (β_0), so the first order Taylor expansion around the initial position become $\sin(\beta) \approx \sin(\beta_0) + \cos(\beta_0)(\beta - \beta_0), \cos(\beta) \approx \cos(\beta_0) - \sin(\beta_0)(\beta - \beta_0)$. Let $\delta\beta = \beta - \beta_0$, the first order approximation of rotation matrix R becomes:

$$
\begin{aligned}
\mathsf{R} &= \mathsf{R}_z(\theta)\mathsf{R}_y(\beta)\mathsf{R}_x(\alpha) \approx \mathsf{R}_z(\theta)\mathsf{R}'_y(\beta)\mathsf{R}'_x(\alpha) \\
&= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\beta_0) - \sin(\beta_0)\delta\beta & 0 & \sin(\beta_0) + \cos(\beta_0)\delta\beta \\ 0 & 1 & 0 \\ -\sin(\beta_0) - \cos(\beta_0)\delta\beta & 0 & \cos(\beta_0) - \sin(\beta_0)\delta\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{bmatrix}
\end{aligned}
$$

Considering the last row of the transformation, the foot position in world frame vertical z-axis becomes

$$
\begin{aligned}
p_{z,j} &= (\mathsf{R}q_j + p_{z,0})_z \\
&\approx -(\delta\beta\cos(\beta_0) - \sin(\beta_0))q_{x,j} + \alpha\cos(\beta_0)q_{y,j} + (\cos(\beta_0) - \delta\beta\sin(\beta_0))q_{z,j} + p_{z,0} \\
&= -\delta\beta(\cos(\beta_0)q_{x,j} + \sin(\beta_0)q_{z,j}) + \alpha\cos(\beta_0)q_{y,j} + (\sin(\beta_0)q_{x,j} + \cos(\beta_0)q_{z,j}) + p_{z,0}
\end{aligned}
$$

Let $h(p_{x,j}, p_{y,j})$ be the stair height at the toe position of jth foot, the relative toe position is $p_{z,j} - h(p_{x,j}, p_{y,j})$. The exact same search algorithm (§3.3) applies with a simple change of coordinates:

$$q'_{x,j} = \cos(\beta_0)q_{x,j} + \sin(\beta_0)q_{z,j}$$
$$q'_{y,j} = \cos(\beta_0)q_{y,j}$$
$$q'_{z,j} = \sin(\beta_0)q_{x,j} + \cos(\beta_0)q_{z,j} - h(p_{x,j}, p_{y,j})$$

The foot velocities in world frame is still $\dot{p}_j = \dot{\mathsf{R}}q_j + \mathsf{R}\dot{q}_j + \dot{p}_0$, so force-moment balance for the planar plane can be solved with a simple change of coordinates $q'_j = \mathsf{R}'_y(\beta)\mathsf{R}'_x(\alpha)q_j$

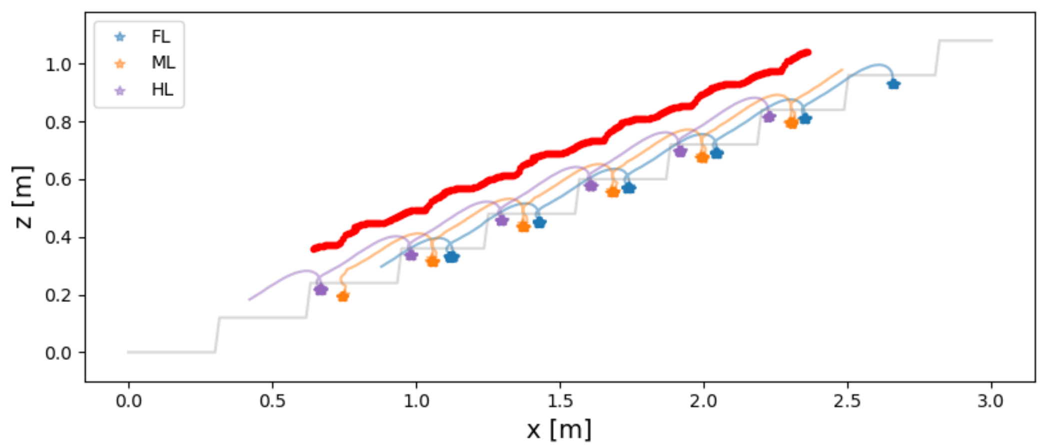

Figure 5.3: A trajectory of simulated BigAnt robot moving up stairs (tread: 0.3m, riser: 0.12m). We plotted the slope (gray), robot center (red), the trajectories of front-left (FL, blue), mid-left (ML, orange), hid-left (HL, purple) feet before compress (stance: stars, swing: lines).

5.3 Residual modeling for sim-to-real transfer

In chapter 3, we presented a reduced-order template model to analyze multi-legged system body motion together with slipping. We showed the model could capture the body motion on BigAnt robot with both tripod and metachronal gait, whereas the model failed to model the side velocities of Multipods (figure 3.11). This might due to robot mechanical asymmetry during installation, or the individual differences in spring stiffness and friction coefficients among all legs. In this section, we assume the locomotion is still geometric, and add a residual term to the model-based mechanical connection matrix to capture the unmodeled behavior.

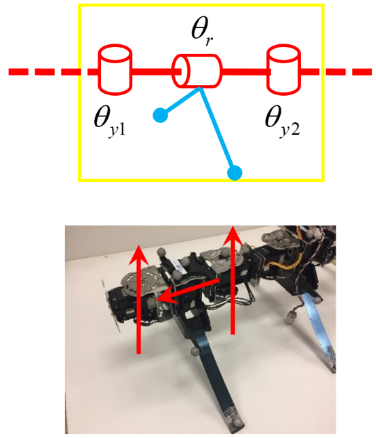

Figure 5.4: The actuation schematic (top) and the modular leg section of Multipod (bottom) in YPY (yaw-roll-yaw) configuration [8].

5.3.1 Residual formulation

Specifically, for an K-legged system, we define a constant matrix $B \in \mathbb{R}^{3\times 2K}$ for the modeling error. Together with $A(q)$ obtained from eqn. (3.20), the geometric reconstruction equation becomes

$$\mathring{g} = \begin{bmatrix} \mathrm{R}_\theta{}^{-1}\dot{p}_{xy,0} \\ \dot{\theta} \end{bmatrix} = (A(q) + B)\dot{q}$$

Let $\mathring{g}_0 = A(q)\dot{q}$ and the ground truth body velocity be $\mathring{g}_{true}$. Let the sub-index t denote the index of snapshots, with N snapshots in total. We obtained matrix B through the least-square regression:

$$B\left[\dot{q}_0, \cdots, \dot{q}_t, \cdots, \dot{q}_N\right] = \left[[\mathring{g}_{true,0} - A(q_0)\dot{q}_0], \cdots [\mathring{g}_{true,t} - A(q_t)\dot{q}_t], \cdots [\mathring{g}_{true,N} - A(q_N)\dot{q}_N]\right]$$
$$B \approx \left[[\mathring{g}_{true,0} - A(q_0)\dot{q}_0], \cdots [\mathring{g}_{true,t} - A(q_t)\dot{q}_t], \cdots [\mathring{g}_{true,N} - A(q_N)\dot{q}_N]\right]\left[\dot{q}_0, \cdots, \dot{q}_t, \cdots, \dot{q}_N\right]^{\dagger}$$

where † represents the Moore-Penrose inverse.

5.3.2 Results

We tested this error formulation against Multipod dataset with 8 legs [58, 59, 60]. Each pair of legs was a leg segment with three motors, configured in yaw-roll-yaw configuration.

The robot gait was defined by

$$\begin{aligned}\theta_{y1} &= \theta_{y1,0}\cos(2\pi f t + i\Delta\phi)\\ \theta_r &= \theta_{r,0}\sin(2\pi f t + i\Delta\phi)\\ \theta_{y2} &= \theta_{y2,0}\cos(2\pi f t + i\Delta\phi)\end{aligned}$$

where $(\theta_{y1}, \theta_r, \theta_{y2})$ were joint angle inputs as shown in the figure 5.4; $(\theta_{y1,0}, \theta_{r,0}, \theta_{y2,0})$ defined the joint amplitudes; f was the gait frequency; $\Delta\phi$ was the phase offset between adjacent leg segments. The gait used roll and yaw amplitude $\theta_{r,0} = 5\pi/36, \theta_{y1,0} = \theta_{y2,0} = \pi/9$, with 26 different phase offset from 0.2π to 1.8π, and 5 gait frequencies $0.3, 0.6, 1.2, 2.4Hz$. Each trial contained robot walking 8-10 cycles, with motion capture sample rate at 125Hz. The detailed gait selection process was documented in chapter 2.3.3.1 [8].

In figure 5.5, we showed the results using four training trials with $\Delta\phi = (0.65, 0.9, 1.1, 1.35)\pi$. We tested the model on a different gait with $\Delta\phi = \pi$. The number of four trials was selected, because the validation error stopped decreasing with increasing number of training trails.

We further studied whether the residual model could generalize from low frequency data to higher frequencies. We used four randomly selected trials running at $0.3Hz$ to estimate B matrix, and tested the body velocity predictions at higher frequencies. We compared the residuals using training trials at it own frequencies. The results were plotted in figure 5.6.

5.3.3 Discussion

We presented a data-driven method to capture the unmodeled dynamics directly from robot motion data. Specifically, we estimated a constant error term for the geometric connection matrix. In figure 5.5, the left and right contours had almost the same error distribution, and both greatly improved the the side velocity estimation. This indicated that the connection residual matrix B could generalize well among different gaits, and frequencies.

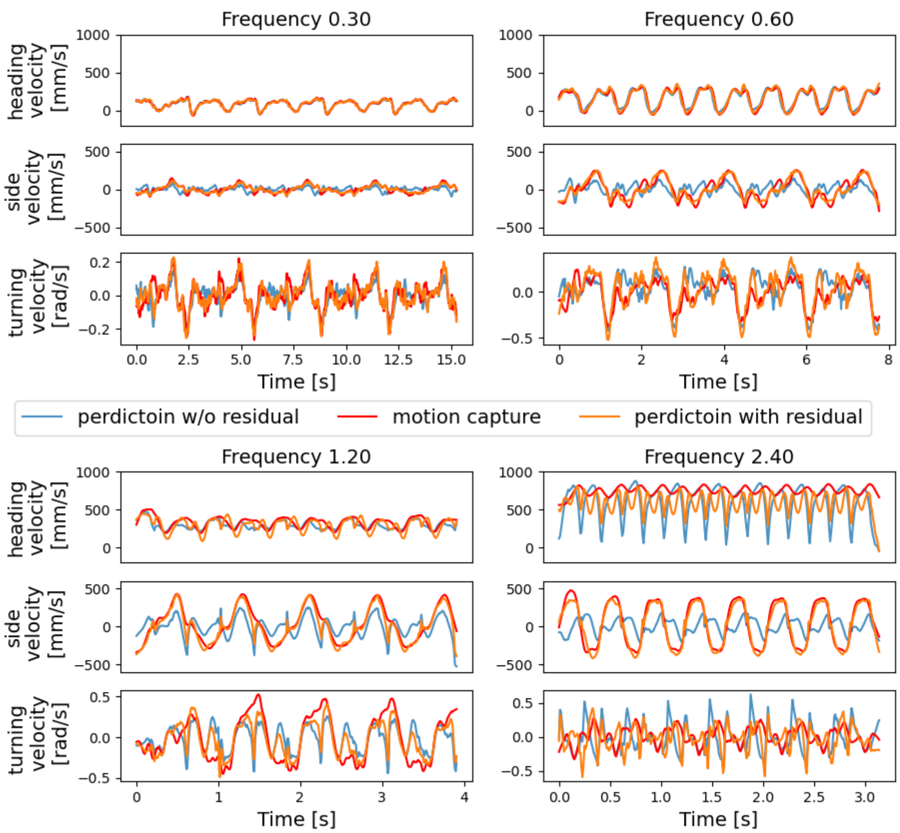

Figure 5.5: We plotted the body velocity prediction with(orange) and without(blue) residual term together with motion capture(red) for Multipod running at 0.3,0.6,1.2,2.4Hz.

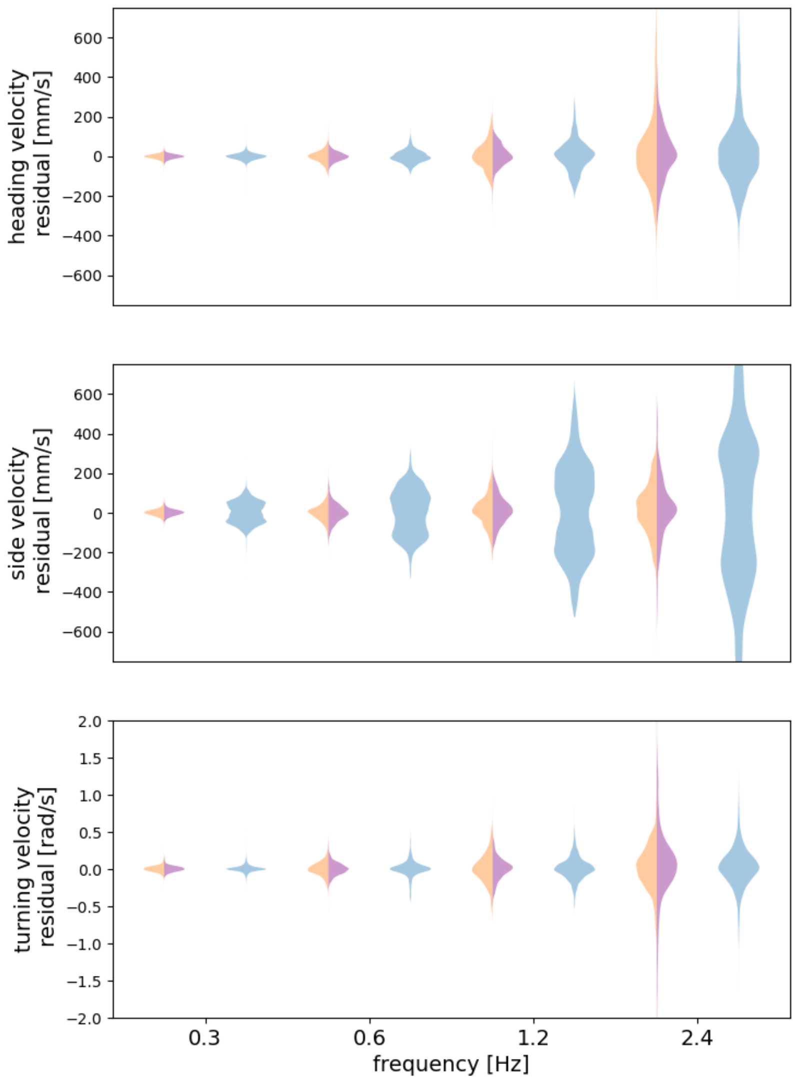

Figure 5.6: Residual plot of model trained on datasets across frequencies. We plotted the violin plots of body velocity residuals of the model (blue), model with residual trained by trials at 0.3Hz (orange, left half), model with residual trained by trials under its own frequency (purple, right half).

5.4 Motion planning with GPU

5.4.1 Acknowledgment

This section is based on joint work with an undergraduate student Shuyu Zhan from University of Michigan.

5.4.2 Introduction

The motion planning problem is always considered as an NP-hard task, and the search could soon become intractable for rather simple disc robots in simple-polygon workspace [89]. Motion planning with contact switches on the legged system is even harder. Many trajectory optimization methods generate motion plans by solving for a nonlinear optimization problem for second-order systems [90, 91]. Alternatively, one could construct asymptotically stable gait primitives for planning [92]. However, with more legs, the planning complexity increases, and the problem could soon become intractable with contact switches over a short planning time horizon. On multi-legged systems, we have shown a history-independent, first-order connection that could model the system well. Since the model could be solved through a simple set of linear equations, we decided to implement this model on GPU. We conducted a brute-force search among a large number of motion plans in parallel on GPU, and select the one with the lowest cost. This allowed us to exhaustively search among the entire configuration space, and generate multi-legged robot motions that are non-intuitive for engineers to design.

5.4.3 Motion planning problem setup

We consider a multi-legged robot with N legs, indexed by $i = 1, \cdots, N$. Each leg has a single periodic degree of freedom (DOF), denoted by $\phi_i \in [0, 1]/\{0, 1\}$. At each timestamp, each DOF, ϕ_i, has M choices of velocity $\dot{\phi}_i^j, j = 1, \cdots, M$. The total choice of the movement at a single timestamp is M^N. The goal of motion planning is, at a given initial condition, to find the best movement of all legs that optimizes a certain goal function. We compute all M^N motion options in parallel on a GPU, and then select the optimal one.

5.4.4 Cost function design

Let $\Omega_t \in SE(2)$ denote the robot position at time step t, and let the i^{th} choice of motion to be $(\dot{\phi})^i$. After executing this motion on the robot for Δt period, the robot transformation becomes Ω_{t+1}^i. Given a target $\Omega_{t+1}^{target} \in SE(2)$, we define the cost function as the distances between the current Ω_{t+1}^i and the target Ω_{t+1}^{target} transformation, as a geodesic norm of translational and rotational

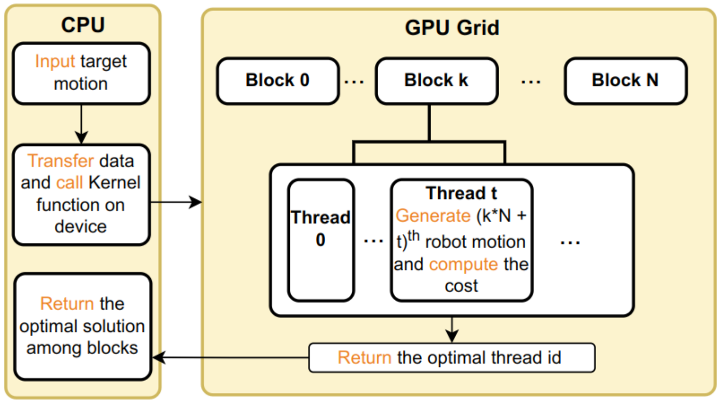

Figure 5.7: Flowchart of GPU parallelization for computing and finding the optimal motion.

distances. Note that different from a cost function that prescribing target velocity, this cost function prescribes a target trajectory, and gives the robot the opportunity to "catch-up" is it deviates from the trajectory.

5.5 Simulation setup

We used the model of 6-legged BigAnt robot in the simulation. Each leg was governed by a four-bar linkage, with foot trajectory shown in figure 5.8. Robot configuration was controlled by motor shaft angles $[\phi_1, \cdots \phi_6]$, with the order [hind left, mid left, front left, hind right, mid right, front right]. At each time step, when computing the robot body velocity given $(\dot{\phi})^i$, we further discretized Δt into T_s finer time steps, and integrated the body velocities to compute the cost. Then we only use the first T_f time steps in the actual motion. This allowed us to look ahead for Δt, without risk of the contact switches causing wrongly predicted motion over this time period. The robot body velocity under certain motor control was computed using the model in §4.2, implemented in Just-in-Time (JIT) compilation offered by Numba [93] in Python for GPU parallelization.

We considered an initial condition of the robot standing with two tripods out of phase, i.e robot configuration $[0.5, 0, 0.5, 0, 0.5, 0]$. We gave each leg four choices of motor velocity ($M = 4$): $\dot{\phi}_i^j \in [-1, 0, 1, 2]$, i.e. move backwards, stay still, move forward or move forward at a higher speed. If a leg was not in contact with the ground, it always had a fixed swinging velocity 2. Therefore, at a given timestamp, we have $M^N = 4^6 = 4096$ many possible motions.

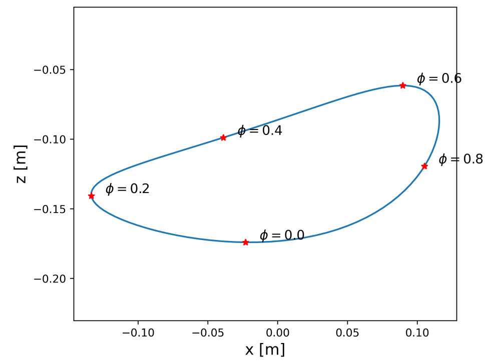

Figure 5.8: Foot trajecotry of Bigant robot. We plotted points at shaft angle $\phi = 0, 0.2, 0.4, 0.6, 0.8$ (red stars).

5.5.1 Results

We tested the planning performance on Nvidia GeForce RTX 1070 and 8 core Intel(R) Core(TM) i7-7700 @ 3.60GHz running Ubuntu 22.04. We set the planning goal for the robot to track a certain constant velocity, and used $\Delta t = 0.05$, $T_s = 20$ and $T_f = 2$.

5.5.1.1 Computation speed

We tested our algorithm on both CPU and GPU. We pre-compiled the functions with JIT. On GPU, we used 5 random goal velocities and each repeated 5 times. The computation speed of planning for 150 time steps cost 1.03 ± 0.05 seconds. Note that the data transfer time between CPU and GPU was measured in this cost. On CPU, the time was quite large, and we tested the planning for 50 time steps, with 5 randomly chosen goal velocities and ran 1 time for each velocity. The CPU time cost was 31 ± 2 seconds per time step. The GPU planning is about 4650 times faster than CPU computation.

5.5.1.2 Planning results

In this section, we showed three example trajectories generated by the planning algorithm. We demonstrated motions of walking straight, turn-in-place, and moving diagonally without turning. Note that the third gaits were quite non-intuitive for BigAnt robot with 1-DoF leg in the xz-plane, and it have never been designed on this robot.

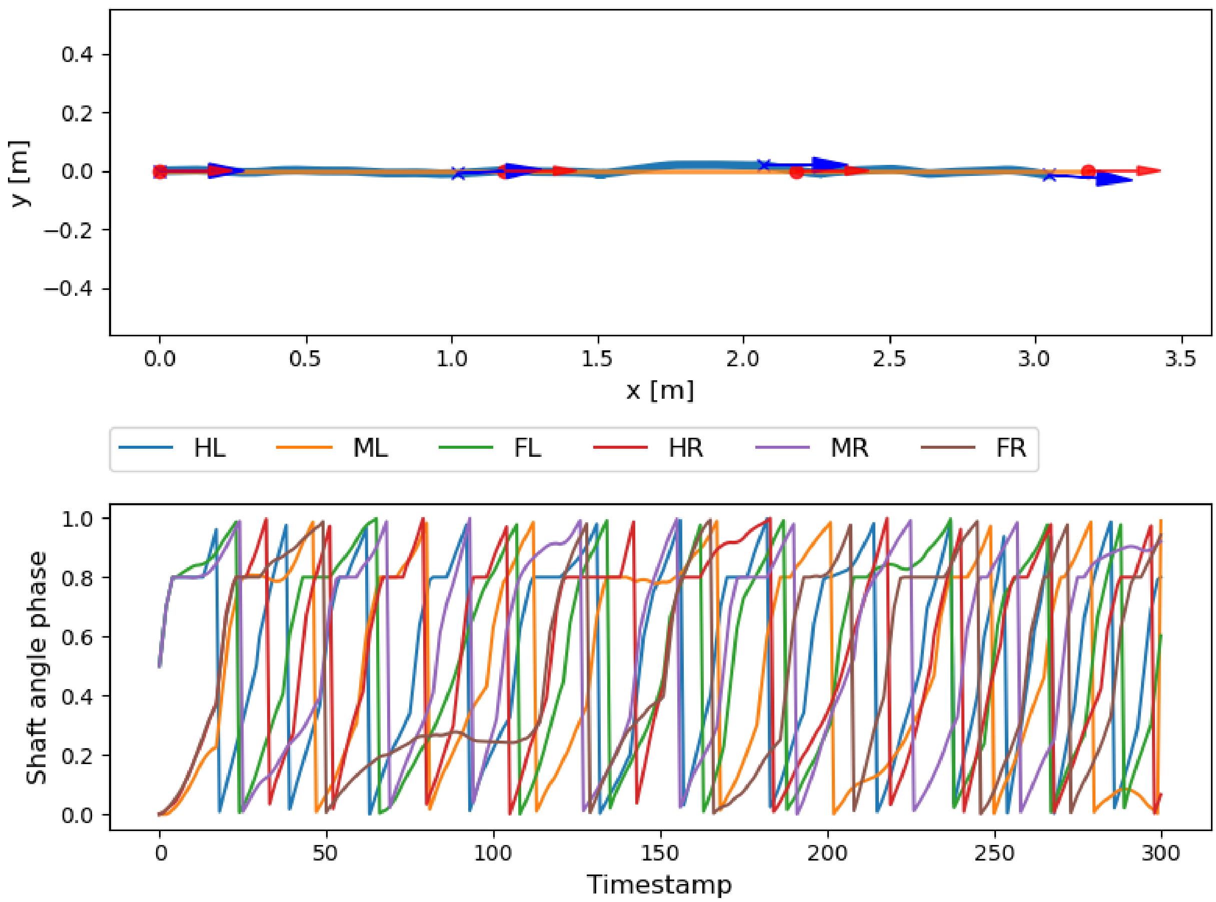

Figure 5.9: We plotted the robot trajectory (top plot) planned (blue) and goal (red), with its yaw orientation at 0, 100, 200, 300 time step ($\Delta t = 0.05$). The goal velocity was 0.2 m/s in x-axis. We also plotted the planned shaft angle (bottom plot) of each leg.

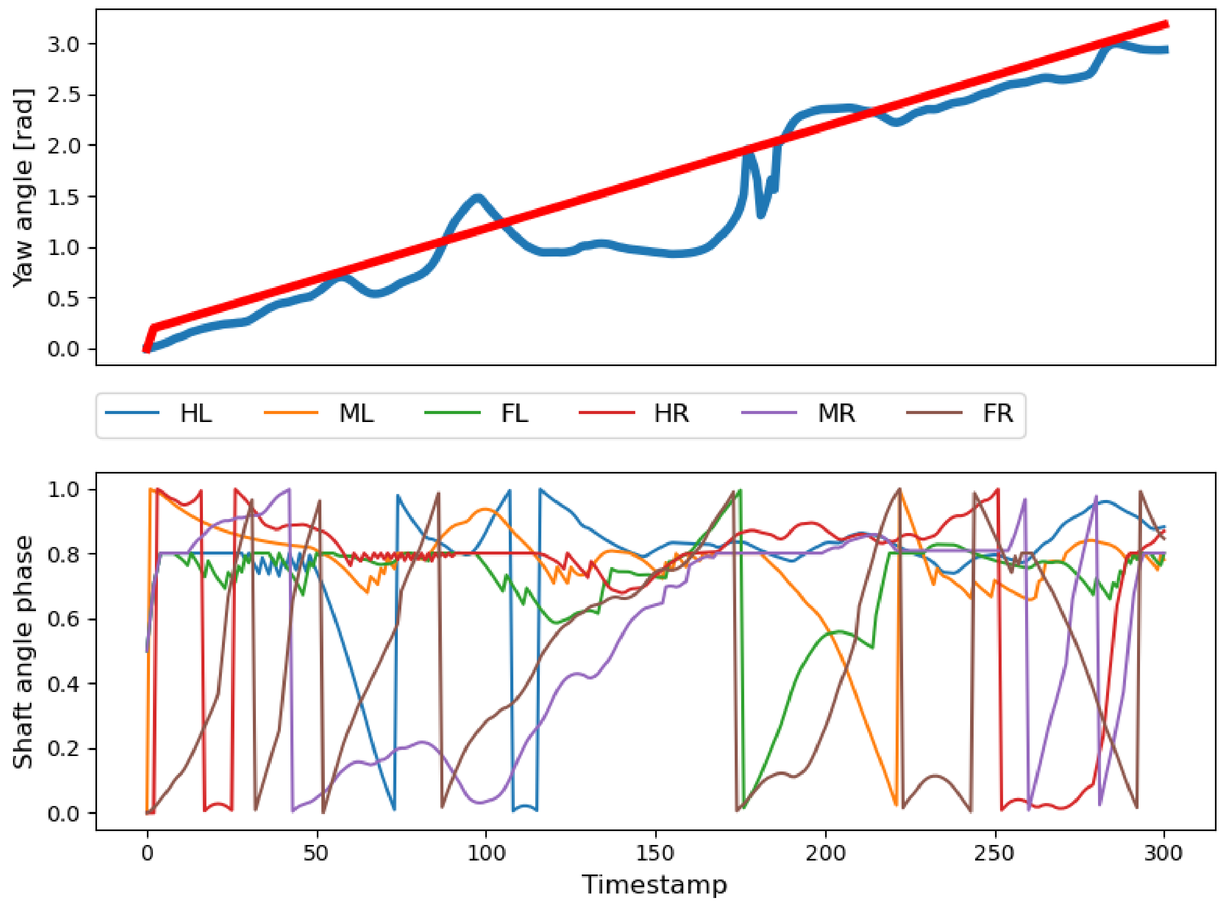

Figure 5.10: We plotted the robot yaw (top plot) planned (blue) and goal (red), $\Delta t = 0.05$. The goal was s turn-in-place motion with 0.2 rad/s in yaw velocity. The planned motion generated displacement always less 0.1m radius around the origin, and thus not shown here. We also plotted the planned shaft angle (bottom plot) of each leg.

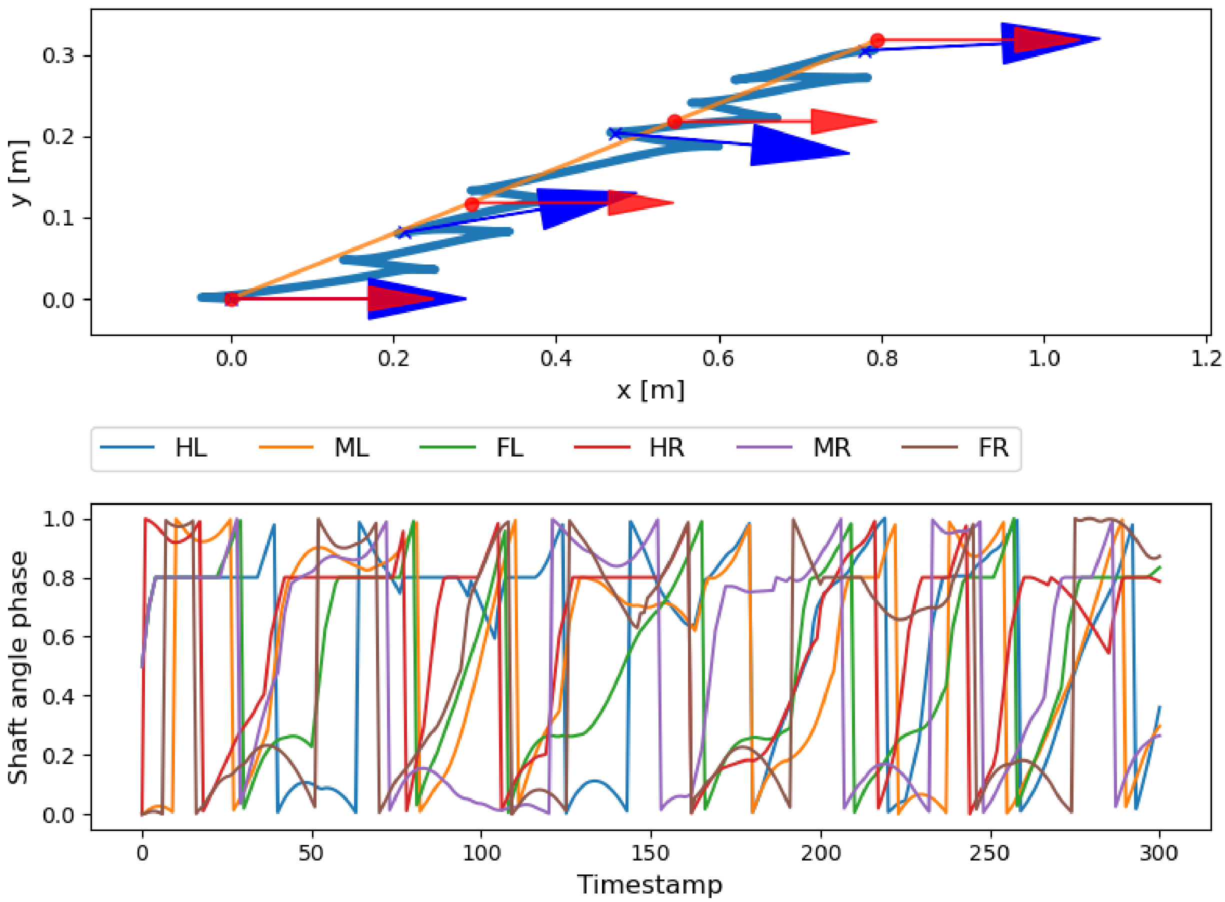

Figure 5.11: We plotted the robot trajectory (top plot) planned (blue) and goal (red), with its yaw orientation at 0, 100, 200, 300 time step (Δt = 0.05). The goal was to generate a side motion, but always pointing the heading to x-axis. The goal velocity was 0.05 m/s in x-axis, and 0.02 in y-axis. We also plotted the planned shaft angle (bottom plot) of each leg.

5.6 Conclusions and discussions

In this section, we presented three extensions and applications of the multi-legged locomotion model proposed in Chapter 3. We extended the model to generalize to special cases of uneven terrain: slopes and staircases. By fitting a residual term in the connection using robot motion data, the model accuracy was improved, and the fitted geometric connection residual motion could be generalized among different gait parameters and frequencies. We also enabled multi-legged motion planning through a simple parallel brute-force search. The motion planning algorithm was fast, with an average computation cost of 300-time steps in 1 second, with 4096 different possibilities considered at each step. It could generate motions that were hard to design by engineers on a hexapod with 1-DoF legs.

5.7 Future work

The examples studied in this section are just a starting point for using this multi-legged locomotion model. There are many more possibilities beyond it. For example, one could mutate the residual modeling together with motion planning to generate motion plans online with feedback. One could also do motion planning on staircases to generate a more robust motion plan of climbing upstairs. We hope this chapter could facilitate the usage of our proposed multi-legged locomotion model, and the development of multi-legged robots with applications in the field.

5.7.1 Anisotropic friction and feet design

Point contact and isotropic friction are two common simplifications in legged robotics models and designs, whereas arthropods usually possess feet with special surface structures to enhance feet attachments [94, 95]. Gecko inspired adhesion has been applied to space manipulation [96] and climbing [97]. In snake locomotion, the anisotropic texture of skins creates drag-direction-dependent friction coefficients that benefit snake slithering with different gaits in their living environments [98, 99]. How anisotropic friction could facilitate multi-legged locomotion remains unclear. In eqn. (3.19), we propose a model to study the effect of friction anisotropy in a certain direction (the ellipsoidal cone in figure 5.12). This proposed model is still symmetrical about the major and minor axis of the ellipse, i.e. the system generates the same friction forces when slipping in two opposite directions. In nature, friction anisotropy can be direction-dependent in the result of an asymmetric arrangement of surface micro-structures [100]. An oblique friction cone that breaks the symmetry of the ellipsoidal cone might be useful to study this tribological property.

Animals also have a variety of feet adapting to their living habitats and lifestyles. For example,

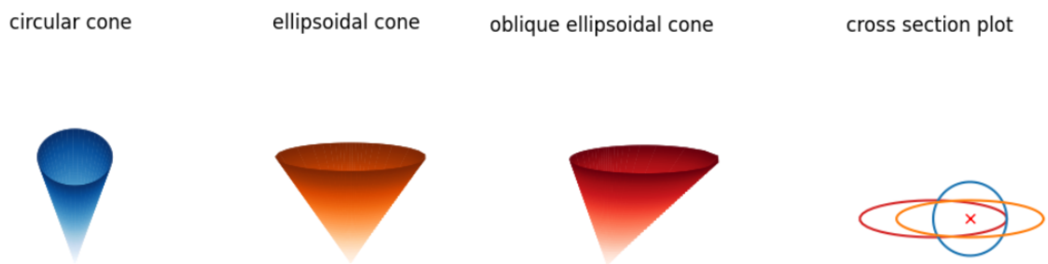

Figure 5.12: Friction cones (from left to right): isotropic friction cone (governed by a single friction coefficient), ellipsoidal friction cone (governed by a friction coefficient together with an enhanced traction direction), oblique ellipsoidal friction cone (governed by a friction coefficient, an enhanced traction direction and center offset), the cross section slice of three cones.

cats have retractable claws with soft pads on their paws; horses have hooves; elephants have large, padded feet with fatty tissue cushions; Kangaroos have large, muscular hind feet with long, strong toes. Current quadrupedal robots usually possess simple elastic round tips as their feet, and some bipedal robots start using human-like feet with ankles to enhance stability. In multi-legged locomotion, future works in feet design considering its shape, friction, and damping together with robot morphology could potentially improve the locomotion robustness and efficiency.

5.7.2 Momentum

In this thesis, we study multi-legged locomotion in the principally kinematic regime, where the robot momentum is negligible and the motion can be modeled through the local connection between body velocity and shape-changing velocity: $g^{-1}\dot{g} = -A(r)\dot{r}$. For a system with non-negligible momentum, the motion written in the full reconstruction equation is [101]:

$$\begin{aligned} g^{-1}\dot{g} &= -A(r)\dot{r} + B(r)p \\ \dot{p} &= \dot{r}^T\alpha(r)\dot{r} + \dot{r}^T\beta(r)p + p^T\gamma(r)p \\ M(r)\ddot{r} &= -C(r,\dot{r}) + N(r,\dot{r},p) + \tau \end{aligned}$$

where p is the generalized momentum. Previous work [29] discussed a data-driven method to model the motion with quickly decaying momentum near period orbit. Our initial results in [102] showed that for large trotting quadrupeds, even though breaking down the assumption of friction annihilates momentum quickly, data-driven connections still approximate their observed motion well. Future work could study what is a good representation of momentum in a legged system, and how to efficiently model it.

Another assumption made in our model was that the shape and shape changing velocities are directly measurable and controllable at a high bandwidth. The robot shape is usually subject to some dynamics in reality, for example, the motor dynamics, or some passive oscillation. The shape and shape velocity $(r, \dot{r})$ becomes not directly controllable, and perhaps also not measurable at a high frequency. How to identify the shape dynamics from robot data could be a future direction to study.

5.7.3 Motion planning

In §5.4, we presented the initial motion planning results on a hexapod. In the search, we picked robot behavior that gave the lowest cost at each time step. This could only give us a local minimum, in the sense that the algorithm was not able to find a path with an accumulated lower cost, when the path needed to deviate from the goal and then come back. Contact planning with many legs is already a high dimensional computationally expensive problem, and how to find the global optimal path can be an interesting direction for future work.

5.7.4 Sensing

State observers usually use proprioceptive information and inertial measurements to estimate robot states. Previous work has shown that an invariant Kalman filter can be used for state estimation over long trajectories [103]. It would be interesting to explore how to integrate our model as part of the system evolution in the filtering scheme. The various system parameters, such as spring constants and friction coefficients, could be converted to slowly varying online estimates, producing an adaptive model. Further utilizing visual sensing in motion planning could be helpful for the motion over complicated uneven terrain.

BIBLIOGRAPHY

[1] Michael LaBarbera. Why the wheels won't go. *The American Naturalist*, 121(3):395–408, 1983.

[2] Samuel A. Burden, Thomas Libby, Kaushik Jayaram, Simon Sponberg, and J. Maxwell Donelan. Why animals can outrun robots. *Science Robotics*, 9(89):eadi9754, 2024.

[3] Neil S. Davies, Russell J. Garwood, William J. McMahon, Joerg W. Schneider, and Anthony P. Shillito. The largest arthropod in Earth history: insights from newly discovered Arthropleura remains (Serpukhovian Stainmore Formation, Northumberland, England). *Journal of the Geological Society*, 179(3):jgs2021–115, 12 2021.

[4] D.M. Gorinevsky and A. Yu. Shneider. Force control in locomotion of legged vehicles over rigid and soft surfaces. *The International Journal of Robotics Research*, 9(2):4–23, 1990.

[5] R. Carlton and S. Bartholet. The evolution of the application of mobile robotics to nuclear facility operations and maintenance. *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, 4:720–726, 1987.

[6] Uluc Saranli, Martin Buehler, and Daniel E. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.

[7] I-Chia Chang, Chih-Hsiang Hsu, Hong-Sheng Wu, and Pei-Chun Lin. An analysis of the rolling dynamics of a hexapod robot using a three-dimensional rolling template. *NONLINEAR DYNAMICS*, 109(2):631–655, JUL 2022.

[8] Dan Zhao. *Locomotion of Low-DoF Multi-legged Robots*. PhD thesis, University of Michigan, 2021.

[9] David Zarrouk, Duncan W. Haldane, and Ronald S. Fearing. Dynamic legged locomotion for palm-size robots. In Thomas George, Achyut K. Dutta, and M. Saif Islam, editors, *Micro- and Nanotechnology Sensors, Systems, and Applications VII*, volume 9467, page 94671S. International Society for Optics and Photonics, SPIE, 2015.

[10] Sangbae Kim, Jonathan E. Clark, and Mark R. Cutkosky. isprawl: Design and tuning for high-speed autonomous open-loop running. *The International Journal of Robotics Research*, 25(9):903–912, 2006.

[11] Roger D Quinn, Gabriel M Nelson, Richard J Bachmann, Daniel A Kingsley, John Offi, and Roy E Ritzmann. Insect designs for improved robot mobility. In *Proceedings of 4th International Conference on Climbing and Walking Robots*, pages 69–76. Germany, 2001.

[12] David Zarrouk and Ronald S. Fearing. Controlled in-plane locomotion of a hexapod using a single actuator. *IEEE Transactions on Robotics*, 31(1):157–167, 2015.

[13] Shinya Aoi, Yoshimasa Egi, and Kazuo Tsuchiya. Instability-based mechanism for body undulations in centipede locomotion. *Phys. Rev. E*, 87:012717, Jan 2013.

[14] Baxi Chong, Yasemin Ozkan Aydin, Jennifer M. Rieser, Guillaume Sartoretti, Tianyu Wang, Julian Whitman, Abdul Kaba, Enes Aydin, Ciera McFarland, Howie Choset, and Daniel I. Goldman. A general locomotion control framework for serially connected multi-legged robots. *CoRR*, abs/2112.00662, 2021.

[15] Dan Zhao Vikram Sachdeva and Shai Revzen. Cockroaches always slip a lot. The Society for Integrative and Comparative Biology Annual Meeting, 2018.

[16] Philip Holmes, Robert J. Full, Dan Koditschek, and John Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Review*, 48(2):207–304, 2006.

[17] R. Blickhan and R. J. Full. Similarity in multilegged locomotion: Bouncing like a monopode. *Journal of Comparative Physiology A*, 173(5):509–517, Nov 1993.

[18] John Schmitt and Philip Holmes. Mechanical models for insect locomotion: dynamics and stability in the horizontal plane i. theory. *Biological Cybernetics*, 83(6):501–515, Nov 2000.

[19] John Schmitt and Philip Holmes. Mechanical models for insect locomotion: dynamics and stability in the horizontal plane – ii. application. *Biological Cybernetics*, 83(6):517–527, Nov 2000.

[20] Ioannis Poulakakis and Jessy 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, 2009.

[21] U. Saranli and D.E. Koditschek. Template based control of hexapedal running. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 1, pages 1374–1379 vol.1, 2003.

[22] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.

[23] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[24] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots, 2021.

[25] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots, 2018.

[26] Ross L Hatton and Howie Choset. Geometric motion planning: The local connection, stokesâĂŹ theorem, and the importance of coordinate choice. *The International Journal of Robotics Research*, 30(8):988–1014, 2011.

[27] Jim Ostrowski and Joel Burdick. The geometric mechanics of undulatory robotic locomotion. *The International Journal of Robotics Research*, 17(7):683–701, 1998.

[28] Brian Bittner, Ross L. Hatton, and Shai Revzen. Geometrically optimal gaits: a data-driven approach. *Nonlinear Dynamics*, 94(3):1933–1948, Nov 2018.

[29] Matthew D. Kvalheim, Brian Bittner, and Shai Revzen. Gait modeling and optimization for the perturbed stokes regime. *Nonlinear Dynamics*, 97(4):2249–2270, Sep 2019.

[30] Dan Zhao, Brian Bittner, Glenna Clifton, Nick Gravish, and Shai Revzen. Walking is like slithering: A unifying, data-driven view of locomotion. *Proceedings of the National Academy of Sciences*, 119(37):e2113222119, 2022.

[31] BIRDS-Lab. Force/torque bigant processed data, 2023.

[32] Ziyou Wu, Dan Zhao, and Shai Revzen. Coulomb Friction Crawling Model Yields Linear ForceâĂŞVelocity Profile. *Journal of Applied Mechanics*, 86(5), 03 2019. 054501.

[33] Gregory L. Wagner and Eric Lauga. Crawling scallop: Friction-based locomotion with one degree of freedom. *Journal of Theoretical Biology*, 324:42–51, 2013.

[34] C. A. Coulomb. Theorie des machines simples. *Memoirs de Mathematique et de Physique de l'Academie Royale.*, 1785.

[35] A. Belyaev et al. *Dynamics of mechanical systems with Coulomb friction*. Springer Science & Business Media, 2012.

[36] Osborne Reynolds. On the theory of lubrication and its application to mr. beauchamp towerâĂŹs experiments, including an experimental determination of the viscosity of olive oil. *Philosophical Transactions of the Royal Society of London*, 177:157–234, IV, 1886.

[37] Henrik Olsson et al. Friction models and friction compensation. *European Journal of Control*, 4(3):176–195, 1998.

[38] R.M. Alexander. The maximum forces exerted by animals. *Journal of Experimental Biology*, 115(1):231–238, 1985.

[39] Rebecca M. Walter and David R. Carrier. Ground forces applied by galloping dogs. *Journal of Experimental Biology*, 210(2):208–216, 2007.

[40] Peter G. Weyand et al. Faster top running speeds are achieved with greater ground forces not more rapid leg movements. *Journal of Applied Physiology*, 89(5):1991–1999, 2000.

[41] Pedro Gregorio, Mojtaba Ahmadi, and Martin Buehler. Design, control, and energetics of an electrically actuated legged robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(4):626–634, 1997.

[42] Giuseppi Gabrielli. What price speed? *Mech. Eng.(ASME)*, 72(10):775–781, 1950.

[43] Andrew J. Clark and Timothy E. Higham. Slipping, sliding and stability: locomotor strategies for overcoming low-friction surfaces. *Journal of Experimental Biology*, 214(8):1369–1378, 04 2011.

[44] V Sachdva, D Zhao, and S Revzen. Cockroaches always slip a lot. In *Integrative and Comparative Biology*, 2018.

[45] R. J. Full and D. E. Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202(23):3325–3332, 12 1999.

[46] Ross Hartley, Maani Ghaffari, Ryan M Eustice, and Jessy W Grizzle. Contact-aided invariant extended kalman filtering for robot state estimation. *The International Journal of Robotics Research*, 39(4):402–430, 2020.

[47] Gerardo Bledt, Patrick M. Wensing, Sam Ingersoll, and Sangbae Kim. Contact model fusion for event-based locomotion in unstructured terrains. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4399–4406, 2018.

[48] Jemin Hwangbo, Carmine Dario Bellicoso, PÃl'ter Fankhauser, and Marco Hutter. Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3872–3878, 2016.

[49] Haldun Komsuoáÿąlu, Kiwon Sohn, Robert J. Full, and Daniel E. Koditschek. A physical model for dynamical arthropod running on level ground. In Oussama Khatib, Vijay Kumar, and George J. Pappas, editors, *Experimental Robotics*, pages 303–317, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[50] Chih-Jan Kao, Chun-Sheng Chen, and Pei-Chun Lin. Reactive force analysis and modulation of the individual legs in a running hexapod robot. In *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 370–375, 2019.

[51] Alfred Shapere and Frank Wilczek. Geometry of self-propulsion at low reynolds number. *Journal of Fluid Mechanics*, 198:557–585, 1989.

[52] Chen Li, Tingnan Zhang, and Daniel I. Goldman. A terradynamics of legged locomotion on granular media. *Science*, 339(6126):1408–1412, 2013.

[53] Tingnan Zhang and Daniel I. Goldman. The effectiveness of resistive force theory in granular locomotiona). *Physics of Fluids*, 26(10):101308, 10 2014.

[54] Baxi Chong, Juntao He, Shengkai Li, Eva Erickson, Kelimar Diaz, Tianyu Wang, Daniel Soto, and Daniel Goldman. Self propulsion via slipping: frictional resistive force theory for multi-legged locomotors. 07 2022.

[55] James R. Usherwood and Benjamin J. H. Smith. The grazing gait, and implications of toppling table geometry for primate footfall sequences. *Biology Letters*, 14(5):20180137, 2018.

[56] Ian Fitzner, Yue Sun, Vikram Sachdeva, and Shai Revzen. Rapidly prototyping robots: Using plates and reinforced flexures. *IEEE Robotics & Automation Magazine*, 24(1):41–47, 2017.

[57] Dan Zhao and Shai Revzen. Multi-legged steering and slipping with low dof hexapod robots. *BIOINSPIRATION & BIOMIMETICS*, 15(4), JUL 2020.

[58] BIRDS-Lab. Birds lab multipod robot motion tracking data - processed data and code, 2021.

[59] BIRDS-Lab. Birds lab multipod robot motion tracking data - raw dataset, 2021.

[60] BIRDS-Lab. Birds lab multipod robot motion tracking data - videos, 2021.

[61] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4397–4404. IEEE, 2015.

[62] Jaemin Yoon, Bukun Son, and Dongjun Lee. Comparative study of physics engines for robot simulation with mechanical interaction. *Applied Sciences*, 13(2), 2023.

[63] Michael Bloesch, Christian Gehring, PÃľter Fankhauser, Marco Hutter, Mark A. Hoepflinger, and Roland Siegwart. State estimation for legged robots on unstable and slippery terrain. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6058–6064, 2013.

[64] Tzu-Yuan Lin, Ray Zhang, Justin Yu, and Maani Ghaffari. Deep multi-modal contact estimation for invariant observer design on quadruped robots. *CoRR*, abs/2106.15713, 2021.

[65] Max Yiye Cao, Stephen Laws, and Ferdinando Rodriguez y Baena. Six-axis force/torque sensors for robotics applications: A review. *IEEE Sensors Journal*, 21(24):27238–27251, 2021.

[66] W. Schwalb, B. Shirinzadeh, and J. Smith. A force-sensing surgical tool with a proximally located force/torque sensor. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 13(1):e1737, 2017.

[67] B. Shimano and B. Roth. On force sensing information and its use in controlling manipulators. *IFAC Proceedings Volumes*, 10(11):119–126, 1977. IFAC International Symposium on Information-Control Problems in Manufacturing Technology, Tokyo, Japan, 17-20 October.

[68] Robert D. Howe. Tactile sensing and control of robotic manipulation. *Advanced Robotics*, 8(3):245–261, 1993.

[69] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Bloesch, H. Kolvenbach, M. Bjelonic, L. Isler, and K. Meyer. Anymal - toward legged robots for harsh environments. *Advanced Robotics*, 31(17):918–931, 2017.

[70] Haldun Komsuoáÿąlu, Kiwon Sohn, Robert J. Full, and Daniel E. Koditschek. A physical model for dynamical arthropod running on level ground. In Oussama Khatib, Vijay Kumar, and George J. Pappas, editors, *Experimental Robotics*, pages 303–317, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[71] X. Alice Wu, Tae Myung Huh, Rudranarayan Mukherjee, and Mark Cutkosky. Integrated ground reaction force sensing and terrain classification for small legged robots. *IEEE Robotics and Automation Letters*, 1(2):1125–1132, 2016.

[72] Pei-Chun Lin, H. Komsuoglu, and D.E. Koditschek. A leg configuration measurement system for full-body pose estimates in a hexapod robot. *IEEE Transactions on Robotics*, 21(3):411–422, 2005.

[73] Jung-Hoon Kim. Multi-axis force-torque sensors for measuring zero-moment point in humanoid robots: A review. *IEEE Sensors Journal*, 20(3):1126–1141, 2020.

[74] Uluc Saranli, Martin Buehler, and Daniel E. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.

[75] Ian Fitzner, Yue Sun, Vikram Sachdeva, and Shai Revzen. Rapidly prototyping robots: Using plates and reinforced flexures. *IEEE Robotics & Automation Magazine*, 24(1):41–47, 2017.

[76] Dan Zhao and Shai Revzen. Multi-legged steering and slipping with low DoF hexapod robots. *Bioinspiration & Biomimetics*, 15(4):045001, may 2020.

[77] Francisco Javier Andrade Chavez, Silvio Traversaro, Daniele Pucci, and Francesco Nori. Model based in situ calibration of six axis force torque sensors. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 422–427, 2016.

[78] ATI Industrial Automation, Apex, NC: USA. *F/T Transducer SixâĂŚAxis Force/Torque Sensor System Installation and Operation Manual*, 2021.

[79] Silvio Traversaro, Daniele Pucci, and Francesco Nori. In situ calibration of six-axis force-torque sensors using accelerometer measurements. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2111–2116, 2015.

[80] Cheng Ding, Yong Han, Wei Du, Jianhua Wu, and Zhenhua Xiong. In situ calibration of six-axis forceâĂŞtorque sensors for industrial robots with tilting base. *IEEE Transactions on Robotics*, pages 1–14, 2021.

[81] Yongqiang Yu, Ran Shi, and Yunjiang Lou. Bias estimation and gravity compensation for wrist-mounted force/torque sensor. *IEEE Sensors Journal*, pages 1–1, 2021.

[82] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. section 3.

[83] R. Courant and D. Hilbert. *Methods of mathematical physics*, volume 1. Interscience publishers, INC., New York, 1937. section VII.§7.2.

[84] Edward Moore and Martin Buehler. Stable stair climbing in a simple hexapod robot. 09 2001.

[85] D. Campbell and M. Buehler. Stair descent in the simple hexapod 'rhex'. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 1, pages 1380–1385 vol.1, 2003.

[86] Arun Bishop and Shai Revzen. Better Stair Climbing By Using Symmetry. In *APS March Meeting Abstracts*, volume 2021 of *APS Meeting Abstracts*, page S14.003, January 2021.

[87] Zhiyi Ren and Aaron Johnson. Toward robust stair climbing of the quadruped using proprioceptive sensing. Technical report, CMU Robotics Institute Summer Scholars Working Papers Journal, 2018.

[88] Carlos Mastalli, Ioannis Havoutis, Michele Focchi, Darwin G. Caldwell, and Claudio Semini. Motion planning for quadrupedal locomotion: Coupled planning, terrain mapping and whole-body control. *CoRR*, abs/2003.05481, 2020.

[89] Paul Spirakis and Chee K. Yap. Strong np-hardness of moving many discs. *Information Processing Letters*, 19(1):55–59, 1984.

[90] Jan Carius, RenÃľ Ranftl, Vladlen Koltun, and Marco Hutter. Trajectory optimization for legged robots with slipping motions. *IEEE Robotics and Automation Letters*, 4(3):3013–3020, 2019.

[91] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.

[92] Robert Gregg, Adam Tilton, Salvatore Candido, Timothy Bretl, and M.W. Spong. Control and planning of 3-d dynamic walking with asymptotically stable gait primitives. *Robotics, IEEE Transactions on*, 28:1415–1423, 12 2012.

[93] *Numba: A llvm-based python jit compiler*, 2015.

[94] Jonas Wolff and Stanislav Gorb. *Attachment Structures and Adhesive Secretions in Arachnids*, volume 7. Springer, 11 2016.

[95] Yu Tian, Noshir S. Pesika, Hongbo Zeng, Kenneth J. Rosenberg, Boxin Zhao, Patricia M. McGuiggan, Kellar Autumn, and Jacob N. Israelachvili. Adhesion and friction in gecko toe attachment and detachment. *Proceedings of the National Academy of Sciences*, 103:19320 – 19325, 2006.

[96] Hao Jiang, Elliot. W. Hawkes, Christine Fuller, Matthew A. Estrada, Srinivasan A. Suresh, Neil Abcouwer, Amy K. Han, Shiquan Wang, Christopher J. Ploch, Aaron Parness, and Mark R. Cutkosky. A robotic device using gecko-inspired adhesives can grasp and manipulate large objects in microgravity. *Science Robotics*, 2(7):eaan4545, 2017.

[97] Daniel Santos, Barrett Heyneman, Sangbae Kim, Noe Esparza, and Mark Cutkosky. Gecko-inspired climbing behaviors on vertical and overhanging surfaces. pages 1125 – 1131, 06 2008.

[98] Jennifer M. Rieser, Tai-De Li, Jessica L. Tingle, Daniel I. Goldman, and Joseph R. Mendelson. Functional consequences of convergently evolved microscopic skin features on snake locomotion. *Proceedings of the National Academy of Sciences*, 118, 2021.

[99] Hamid Marvi and David L. Hu. Friction enhancement in concertina locomotion of snakes. *Journal of The Royal Society Interface*, 9:3067 – 3080, 2012.

[100] Halvor T. Tramsen, Stanislav N. Gorb, Hao Zhang, Poramate Manoonpong, Zhendong Dai, and Lars Heepe. Inversion of friction anisotropy in a bio-inspired asymmetrically structured surface. *Journal of The Royal Society Interface*, 15, 2017.

[101] Anthony M. Bloch, Perinkulam S. Krishnaprasad, Jerrold E. Marsden, and Richard M. Murray. Nonholonomic mechanical systems with symmetry. *Archive for Rational Mechanics and Analysis*, 136:21–99, 1996.

[102] Ziyou Wu, Shai Revzen, and Jeffrey Duperret from Ghost Robotics Team. Connection-Based Data-Driven Gait Modeling of a Quadruped. In *APS March Meeting Abstracts*, volume 2023 of *APS Meeting Abstracts*, page W10.009, January 2023.

[103] Tzu-Yuan Lin, Tingjun Li, Wenzhe Tong, and Maani Ghaffari. Proprioceptive invariant robot state estimation, 2024.