Data-Augmented Motion Planning and Predictive Control for Robotic Jumping on Highly Dynamical Terrains

by

Yuhao Huang

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science

(Mechanical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2025

Date of final oral examination: 07/05/2025

The dissertation is approved by the following members of the Final Oral Committee:
Xiaobin Xiong, Assistant Professor, Mechanical Engineering
Peter Adamczyk, Associate Professor, Mechanical Engineering
Jeremy Coulson, Assistant Professor, Electrical Engineering

acknowledgments

I am deeply grateful to the individuals who provided support, valuable feedback, and companionship throughout my academic journey.

First of all, I would like to express my sincere gratitude to my master's supervisor, Prof. Xiaobin Xiong, for welcoming me into this wonderful lab, providing abundant resources, and fostering a supportive environment with incredible labmates. His frequent and sincere guidance helped me develop a deep understanding of the fundamental principles of research. I truly respect his thoughtfulness and steadfast attitude toward research, and I will continue to carry this mindset forward in my future studies.

Next, I would like to express my heartfelt gratitude to my incredibly supportive current and former labmates: Jiarong Kang, Yicheng Zeng, Yi Wang, Hanwen Wang, Arturo Gamboa-Gonzalez, and Yunfu Deng. We have spent countless hours discussing and debating research ideas and directions, and without your time, insights, and support, I would not have made it through this journey. I would like to offer my deepest thanks to Jiarong for his invaluable help with formula derivations and experiments for various papers and my thesis. His calm and open-minded communication style has left a lasting impression on me, and I will carry it with me throughout my future endeavors. I would also like to express my sincere gratitude to Yicheng Zeng, who collaborated closely with me on several publications. His insightful understanding of various topics has been invaluable to our papers and this thesis.

I would also like to express my heartfelt gratitude to my family—my parents, grandparents, aunt, uncle, and younger sister—as well as my girlfriend. Their unwavering emotional support has been essential in sustaining me throughout this long journey.

Finally, I would like to thank anyone who has helped me along the way during my master's journey.

abstract

Model-based controllers have shown strong performance in a wide range of robotic systems, from relatively simple platforms such as drones and unicycle cars to more complex systems such as quadrupedal and bipedal robots. These controllers also provide a principled framework for synthesizing diverse behaviors in robotic systems, including jumping in hopping robots, trotting, pacing, and bounding in quadrupeds, and walking or running in humanoid robots.

At the core of model-based control lies the use of dynamical models to predict future robot behavior based on current states and control inputs. However, discrepancies between the idealized models used for control and the real-world physical systems are often a major source of control failure. These mismatches can stem from various sources, most commonly from inaccurately modeled dynamics, such as rigid-body assumptions that ignore structural deformation and unmodeled dynamics, including compliance in robot-environment interactions (REI) and oversimplified representations of internal component behavior.

To address these challenges, this thesis proposes a Data-Augmented Control (DAC) framework that leverages input-output data to model complex or qualitatively hard to define dynamics and integrates them into both motion planning and predictive control. By directly incorporating fitted interaction dynamics into the control pipeline, the DAC framework enables more dynamic, and efficient locomotion behaviors, particularly in scenarios involving compliant or unstructured environments.

contents

Acknowledgments i

Abstract ii

Contents iii

List of Figures iv

1 Introduction 1

2 Preliminaries 4
- *2.1 Williems' Fundamental Lemma* 4
- *2.2 Data-enabled Predictive Control (DeePC)* 7
- *2.3 Slacked PEM-MPC Formulation of DeePC* 9
- *2.4 Control of PogoX for Flying-Hopping Locomotion* 11

3 Data-Augmented Control of PogoX Hopping on Dynamic Terrains 16
- *3.1 Data-Augmented Interaction Dynamics* 17
- *3.2 Data-Augmented Motion Planning* 19
- *3.3 Optimization-Based State Estimation* 23
- *3.4 Online Data-Augmented Predictive Control (DAPC)* 24
- *3.5 Results* 28

4 Summary 33

References 34

ivst of

2.1 The mechanical components of PogoX. 12
2.2 Hardware components of PogoX. (figure converted from Wang et al. (2024)) 13
2.3 Hopping control framework of PogoX (figure converted from Wang et al. (2024)) 15
2.4 Overall behavior of flying-hopping locomotion (figure converted from Wang et al. (2024)) 15

3.1 General Framework of DAC. 16
3.2 Demonstration of input (gray) output (green) of data augmented dynamics in flight phase (blue) and ground phase (red). 18
3.3 Plot of predicted trajectories (red) versus actual trajectories (blue). 29
3.4 Plot of trajectory tracking results in simulation. 31
3.5 Results of hardware air mattress hopping: (a) demonstrates the setups of hardware experiment. (b) is the plot of trajectory tracking results on hardware. 32

1 introduction

Model-based control methods lie at the core of modern robotics and industrial applications. Their successful deployment across a wide range of robotic systems, including quadrotors such as those in Mellinger and Kumar (2011), Sreenath et al. (2013), Brescianini (2013), Sun et al. (2022), manipulators such as those in Lynch and Park (2017), Russ (2024), Murray et al. (1994), and legged robots such as those in Di Carlo et al. (2018), Russ (2023),Neunert et al. (2018), highlights their critical role for robotics researches. These methods enable the generation of agile yet precise, dynamics-aware locomotion behaviors that consider interactions with the environment, making them an indispensable component in achieving full autonomy for robotic systems.

Models lie at the core of model-based control methods. Extensive literature on system identification, such as Wensing et al. (2024), Khalil and Bennis (1995), and Simpkins (2012), has demonstrated the importance of accurate models for both fixed-base and floating-base robotic systems. Various techniques have been developed to address challenges such as inertia parameter identification for floating-base systems, handling closed kinematic chains, and dealing with noise in real sensor measurements. A comprehensive overview of these topics can be found in Leboutet et al. (2021).

Robot-Environment Interaction (REI) is a fascinating yet challenging research area, largely due to the lack of precise and computationally efficient models for compliant contact behaviors described in Le Lidec et al. (2024). Rigid-body-dynamics-based contact modeling offers a theoretically rigorous approach to describing contact phenomena. However, the rigid-body assumption neglects the inherent compliance present in real-world interactions, limiting its applicability in many scenarios. To address this, some researchers have adopted static compliant models described

in Marhefka and Orin (1999), Xie et al. (2022) and Khadiv et al. (2019), which provide continuous and more tractable formulations that better reflect the compliant nature of physical contact. However, these models are inherently static and fail to capture the dynamic characteristics of contact interactions. In highly dynamic scenarios, such as rapid locomotion over deformable or unpredictable terrains, contact behavior is significantly influenced by past state transitions, which static models are unable to represent. As a result, there remains a critical need in the robotics and control communities for dynamical contact models that are both compliant in nature and computationally efficient, enabling real-time application in dynamic environments.

There is extensive literature on identifying unknown system dynamics using data. Traditional system identification approaches aim to improve the accuracy of parametric model parameters through data, thereby enhancing the performance of model-based controllers. However, these methods often require substantial domain expertise, and the collected data is typically not integrated directly into the control loop. This limits the ability to fully exploit and update data in real-time control scenarios. In contrast, Data-Enabled Predictive Control (DeePC), a data-driven control framework based on non-parametric models, introduced in Coulson et al. (2018), has recently gained significant attention for its capability to directly incorporate sensor data into predictive control formulations. By removing the need for an explicit system model, DeePC shifts the modeling burden onto the data itself—enabling the optimization process to implicitly determine a system representation that best fits the observed behavior.

This method has been shown to be effective across various robotic systems, including quadrotors in Elokda et al. (2021), Coulson et al. (2018), manipulators in Carron et al. (2019), and excavators in Wegner (2021). Its ability to directly generate control inputs from sensor measurements or

state estimation outputs makes it particularly well-suited for real-world control scenarios involving approximately linear robotic systems.

In this thesis, we propose a Data-Augmented Control (DAC) framework to enable real-time control of PogoX, a self-designed flying-hopping robot that is inherently underactuated and exhibits hybrid dynamics. The framework is designed to achieve highly dynamic jumping motions on dynamic terrains such as trampolines and air mattresses by actively incorporating the interaction dynamics into the planning and control formulation through data-driven modeling.

Our key contributions are as follows:

- We propose a novel Data-Augmented Control (DAC) framework that actively models and leverages interaction dynamics to improve control performance.
- We introduce a Data-Augmented Motion Planning (DAMP) framework for generating trajectories that are physically consistent with the interaction dynamics.
- We develop a Data-Augmented Predictive Control (DAPC) framework that integrates interaction dynamics into the predictive control formulation, enabling data-driven adaptation to complex physical interactions.

2 preliminaries

In this chapter, we provide a brief introduction to the mathematical principles underlying the Data-enabled Predictive Control (DeePC) framework, along with the control framework of PogoX, which serves as the primary testbed for the proposed control methods. We begin by reviewing the mathematical formulation of Willems' Fundamental Lemma, which serves as the theoretical foundation of DeePC. Next, we introduce the slacked Prediction Error Method Model Predictive Control (PEM-MPC) formulation, designed to improve noise robustness and computational efficiency of DeePC. Finally, we present the complete control framework for the flying-hopping locomotion of PogoX.

2.1 Williems' Fundamental Lemma

A discrete linear time-invariant (LTI) system is a type of dynamical systems whose behavior can be described by the following equations:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{2.1}$$

where $A \in R^{n\times n}$, $B \in B^{n\times m}$, $C \in R^{p\times n}$, $D \in R^{p\times m}$ and $q = m + p$.

Willems' Fundamental Lemma, primarily developed by Willems et al. (2005), states that an input-output trajectory collected from a discrete linear time-invariant (LTI) system, given that the input trajectory is persistently excited, can be used to parameterize all possible trajectories that the system can generate. Next, we present the detailed mathematical definitions and formulations of Williems' Fundamental Lemma that will be utilized in the subsequent sections.

Definition 2.1. *Behavior of a discrete time LTI system 2.1 is:*

$$\mathsf{B} := \{w = (u, y) : Z_{\geqslant 0} \to R^{q} \mid \exists x \in Z_{\geqslant 0},\ \textit{s.t. LTI holds}\ \forall t \in Z_{\geqslant 0}\}. \tag{2.2}$$

An element of the behavior is referred to as an *input-output trajectory*. Consider a pair (u, x, y) where u is the control input, x is the state vector and y is the output the LTI 2.1, then the pair is called *input-state-output* trajectory. A collection of input-output trajectories over a specific time interval defines the *restricted behavior* of the system.

Definition 2.2. *Restricted behavior of a discrete time LTI system 2.1 is:*

$$\mathsf{B}|_{[t_0,t_1]} = \{W_{[t_0,t_1]} | w \in \mathsf{B}\}, \tag{2.3}$$

$$\textit{where}\ W_{[t_0,t_1]} := \begin{bmatrix} w(t_0) \\ \vdots \\ w(t_1) \end{bmatrix} \in R^{q(t_1-t_0+1)},\ t_0, t_1 \in Z_{\geqslant 0}\ \ \textit{with}\ \ t_0 \leqslant t_1.$$

Given a sequence of input-output trajectories $W_{[t_0,t_1]}$, the *Hankel matrix* of depth L is defined as below:

$$\mathsf{H}_L(W_{[t_0,t_1]}) := \begin{bmatrix} w(t_0) & w(t_0+1) & \cdots & w(t_1-l+1) \\ w(t_0+1) & w(t_0+2) & \cdots & w(t_1-l+2) \\ \vdots & \vdots & \cdots & \vdots \\ w(t_0+l-1) & w(t_0+l) & \cdots & w(t_1) \end{bmatrix} \tag{2.4}$$

where $\mathsf{H}_L \in R^{L} \times R^{t_1-t_0-L+2}$. With the definition of the Hankel matrix, we can now introduce the concept of *persistent excitation*.

Definition 2.3. *Let* $u : Z_{\geqslant 0} \to R^{m}$. *Let* $t_0, t_1 \in_{\geqslant 0}$ *with* $t_0 \leqslant t_1$. *Let* $L \leqslant (t_1 - t_0 - 1)$. *Then a sequence of input trajectories* $u_{[t_0,t_1]}$ *is said to be persistently*

exciting of order L *if its Hankel matrix* $H_L(u_{[t_0,t_1]})$ *has full rank:*

$$\mathit{rank}(H_L(u_{[t_0,t_1]})) = mL. \tag{2.5}$$

Alsalti et al. (2023)

With all the pieces needed, we now state the definition of Williems' Fundamental Lemma.

Theorem 2.4. *For an LTI system 2.1 and assume* (A, B) *is controllable. Let* $T, L \in \mathbb{Z}_{\geqslant 0}$*. Let* $(u_{[0,T-1]}, x_{[0,T-1]}, y_{[0,T-1]})$ *be an input-state-output trajectory. If* $u_{[0,T-1]}$ *is persistently exciting of order* $L + n$ *then:*

$$\begin{aligned} &1.\quad \mathit{rank}\begin{pmatrix} H_1(x_{[0,T-L]}) \\ H_L(u_{[0,T-1]}) \end{pmatrix} = mL + n, \\ &2.\quad \mathcal{B}|_{[0,L-1]} = \mathit{Im}\begin{pmatrix} H_L(u_{[0,T-1]}) \\ H_L(y_{[0,T-1]}) \end{pmatrix}, \end{aligned} \tag{2.6}$$

where Im means column space.

The theorem above gives us the corollary below which forms the mathematical basis of DeePC.

Corollary 2.5 (Willems' Fundamental Lemma)**.** *Consider the LTI system and assume* (A, B) *is controllable. Let* $T, L \in \mathbb{Z}_{\geqslant 0}$*. Let* $(u_{[0,T-1]}, y_{[0,T-1]}) \in \mathcal{B}|_{[0,T-1]}$*. If* $u_{[0,T-1]}$ *is persistently exciting of order* $L + n$ *then for a sequence of newly collected data* $(\bar{u}_{[0,L-1]}, \bar{y}_{[0,L-1]})$*, it satisfies* $(\bar{u}_{[0,L-1]}, \bar{y}_{[0,L-1]}) \in \mathcal{B}|_{[0,L-1]}$ *if and only if there exists* g *such that:*

$$\begin{pmatrix} H_L(u_{[0,T-1]}) \\ H_L(y_{[0,T-1]}) \end{pmatrix} g = \begin{pmatrix} \bar{u}_{[0,L-1]} \\ y_{[0,L-1]} \end{pmatrix}. \tag{2.7}$$

The corollary states that all sequences of input-output trajectories of an LTI system 2.1 must lie within the image of Hankel matrices of depth L, constructed from a collected sequence of input-output trajectories.

2.2 Data-enabled Predictive Control (DeePC)

DeePC is a direct application of Willems' fundamental lemma within the Model Predictive Control (MPC) framework. To apply Williems' fundamental lemma to a predictive control framework, we must be able to roll out the system dynamics. This means that, given an initial condition and an input sequence $u_{[0,T_f-1]}$ of length $T_f \in \mathbb{Z}_{\geqslant 0}$, we need to determine the corresponding output response $y_{[0,T_f-1]}$ for the LTI system 2.1, based on the input-output dynamics.

The statement is equivalent to: given input-output data $(u_{[0,T-1]}, y_{[0,T-1]}) \in \mathcal{B}|_{[0,T-1]}$, an initial trajectory $(u_{ini}, y_{ini}) \in \mathcal{B}|_{[0,T_{ini}-1]}$ and a future input sequence $\bar{u}_{[0,T_f-1]} \in \mathbb{R}^{mT_f}$, we want to find $\bar{y}_{[0,T_f-1]}$ such that:

$$\begin{bmatrix} u_{ini} \\ \bar{u}_{[0,T_f-1]} \\ y_{ini} \\ \bar{y}_{[0,T_f-1]} \end{bmatrix} \in \mathcal{B}|_{[0,T_{ini}+T_f-1]}. \tag{2.8}$$

One obvious problem regarding this control problem is that the output response may not be uniquely identified given that the input is not long enough. To quantify the length of input sequence required to produce a unique output sequence, we define *lag* of LTI 2.1 system.

Definition 2.6. *The lag of an LTI system is defined as the smallest integer* k *such that the corresponding observability matrix* $\mathcal{O}_k$ *has rank* n*. Lag is denoted as* $\ell(\mathcal{B})$*.*

Then it can be proved that if T_{ini} is bigger than the lag of LTI, there is a unique initial condition x_0 to the input-state-output trajectory and thus the corresponding output sequence is unique.

Lemma 2.7. *Let* $T_{ini} \geqslant \ell(B)$ *and let* $\begin{pmatrix} u \\ y \end{pmatrix} \in B|_{[0,T_{ini}+T_f-1]}$. *Then there exists a unique* $x_{ini} \in R^n$ *such that*

$$y = O_{T_f} x_{ini} + T_{T_f} u, \tag{2.9}$$

where T_{T_f} *is the Toeplitz matrix. (Coulson et al. (2018))*

Having established the necessary restrictions on the input sequence length, we can now present the formulation of DeePC (Coulson et al. (2018)).

Definition 2.8 (Data-Enabled Predictive Control)**.** *Let* T_f *be the prediction horizon,* $T_{ini} \geqslant \ell(B)$ *as the initial trajectory length,* $U \subseteq R^{mT_f}$ *as the input constraint set and* $Y \subseteq R^{pT_f}$ *be the output constraint set. DeePC is an optimal control problem formulated as:*

$$\begin{aligned} \min_{g,u,y} \quad & \sum_{j=0}^{T_f-1} \|y(j) - y^{ref}(j)\|_Q^2 + \|u(j) - u^{ref}(j)\|_R^2 + \lambda_g \|g\|^2 \\ s.t. \quad & \begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{pmatrix}, \\ & u \in U, \\ & y \in Y, \end{aligned} \tag{2.10}$$

where $Q \succeq 0, R \succ 0$ *and* Q *subscript under the Mahalanobis norm* $\|\cdot\|$ *represents information matrix of each term,* $\lambda_g \in \mathbb{R}$ *is a tunable gain.*

2.2.1 Robust DeePC

Since data collected from real hardware are often noisy and sometimes corrupted, there are scenarios where a given initial trajectory y_{ini}, obtained from direct sensor readings or state estimation, may lead to an infeasible optimal control problem 2.8. To address this, we follow the formulations in Huang et al. (2023) and Didier et al. (2021) to incorporate noise robustness into our framework by incorporating a slack variable σ_y into the formulation:

$$
\begin{aligned}
\min_{g,u,y} \quad & \sum_{j=0}^{T_f-1} \|y(j) - y^{ref}(j)\|_Q^2 + \|u(j) - u^{ref}(j)\|_R^2 + \lambda_g\|g\|^2 + \lambda_y\|\sigma_y\|^2 \\
\text{s.t.} \quad & \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} + \sigma_y \\ u \\ y \end{bmatrix} \\
& u \in \mathcal{U}, \\
& y \in \mathcal{Y},
\end{aligned}
\tag{2.11}
$$

where $\lambda_g, \lambda_y \in \mathbb{R}$ are tunable gains that control the penalization on linear predictor and the noise slackness.

2.3 Slacked PEM-MPC Formulation of DeePC

In Eqn. 2.8, it can be observed that the size of the optimization problem is strongly dependent on the dimension of the decision variable g, given an initial trajectory length T_{ini} and a prediction horizon T_f. Since the

input data need to be persistently exciting of order $T_{ini} + T_f + n$, ensuring that rank $H_{T_{ini}+T_f+n}(u_{[0,T-1]}) = m(T_{ini} + T_f + n)$, the Hankel matrix $H_{T_{ini}+T_f+n}(u_{[0,T-1]})$ must have at least many columns as many rows. Consequently, the size of g is at least $m(T_{ini} + T_f) + (m + 1)n$.

However, since Willems' fundamental lemma is theoretically exact only for LTI systems, applying it to systems that are approximately linear but inherently nonlinear requires a significantly larger dataset, as well as longer initial trajectory length and prediction horizon, for practical effectiveness. As a result, the size of g increases substantially, posing a major challenge for the real-time implementation of DeePC in complex dynamical systems.

In this thesis, we build upon and enhance the strategies proposed in Huang et al. (2019) and Fawcett et al. (2022) by incorporating slack variables into the PEM-MPC formulation to improve noise robustness while preserving computational efficiency.

The slacked PEM-MPC formulation of DeePC is formulated to first finding an approximation of g by formulating the following quadratic programming problem:

$$\begin{aligned} \min_{g,\sigma_y} \quad & \|g\|_Q^2 + \|\sigma_y\|_R^2 \\ \text{s.t.} \quad & \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} + \sigma_y \\ u \end{bmatrix} \end{aligned} \tag{2.12}$$

Since this problem is an equality constrained quadratic program, it can be solved by directly solving its Karush-Kuhn-Tucker (KKT) optimality condition. Here we let $A = \begin{bmatrix} U_p & 0 \\ Y_p & -I \\ U_f & 0 \end{bmatrix}$, $G = \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix}$ and $b = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \end{bmatrix}$

then the solution can be written as:

$$\begin{pmatrix} g \\ \sigma_y \end{pmatrix} = G^{-1}A^T(AG^{-1}A^T)^{-1}b. \tag{2.13}$$

Here we define $B = \begin{pmatrix} Y_f & 0 \\ 0 & I \end{pmatrix}$ and define data transition matrix as: $G_{slack} = BG^{-1}A^T(AG^{-1}A^T)^{-1}$. Then the slacked PEM-MPC formulation can be described as below.

Definition 2.9 (Slacked PEM-MPC formulation of DeePC)**.**

$$\begin{aligned} &\min_{u,y,\sigma_y} \|y - y_{des}\|_Q^2 + \|u - u_{des}\|_R^2 + \|\sigma_y\|_P^2 \\ &\text{s.t.} \quad \begin{pmatrix} y \\ \sigma_y \end{pmatrix} = G_{slack} \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \end{pmatrix}, \end{aligned} \tag{2.14}$$

where G *is called the Data-Transition Matrix.*

A more detailed description of this formulation can be found in our paper Zeng et al. (2024).

2.4 Control of PogoX for Flying-Hopping Locomotion

In this section we present the primary control framework of PogoX which is a customized flying-hopping robot designed by our lab. The primary design ideas can be found in Wang et al. (2024). It consists of a quadroter for effective flying and a pogo stick attached to the base of the quadroter for dynamic hopping as shown in Fig. 2.1.

The current hardware design of the robot can be found in Kang et al. (2024). The robot's computational resources consist of a Raspberry Pi 4B

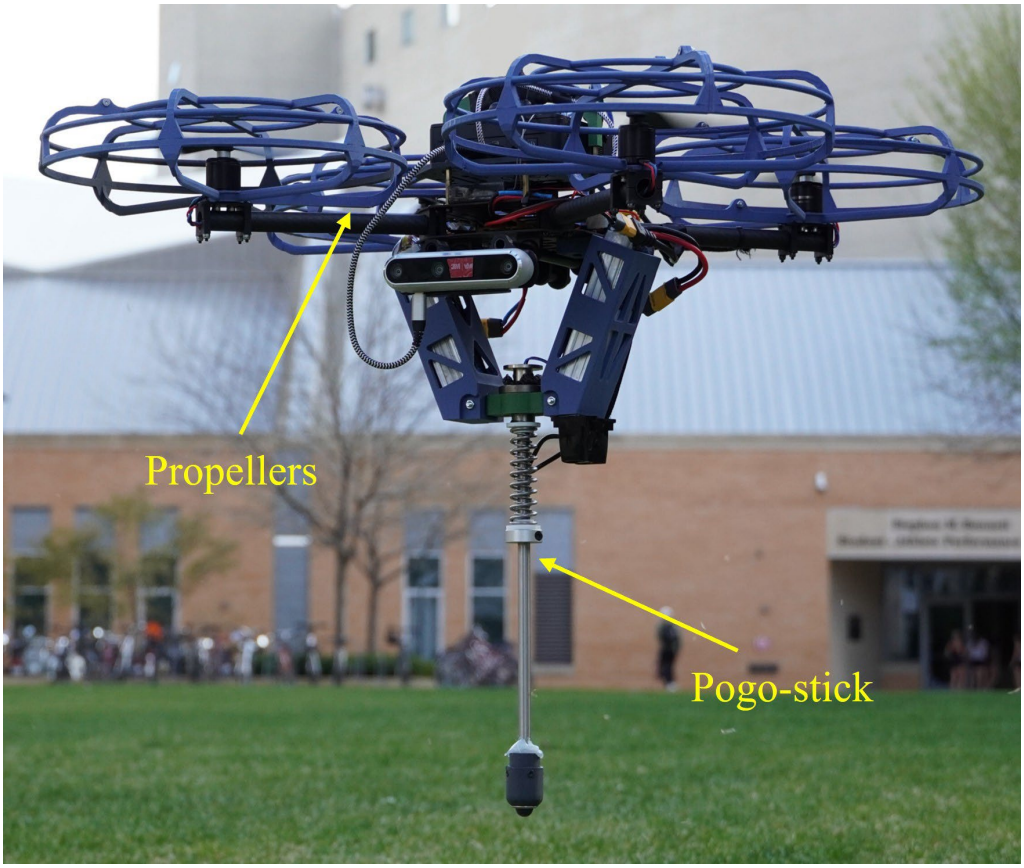

Figure 2.1: The mechanical components of PogoX.

for contact sensor and 1d lidar interfacing, and an Intel NUC for state estimation and control. An onboard IMU is integrated with the flying controller PX4 for raw orientation estimation and its readings are sent to Intel NUC through USB. Four brushless DC motors are controlled by PX4, which sends PWM commands to the ESC motor controllers to precisely regulate motor speed. All electronic components are powered by two 4S LiPo batteries, and the targets are controlled via a remote controller (RC). The complete hardware configuration is illustrated in Fig. 2.2.

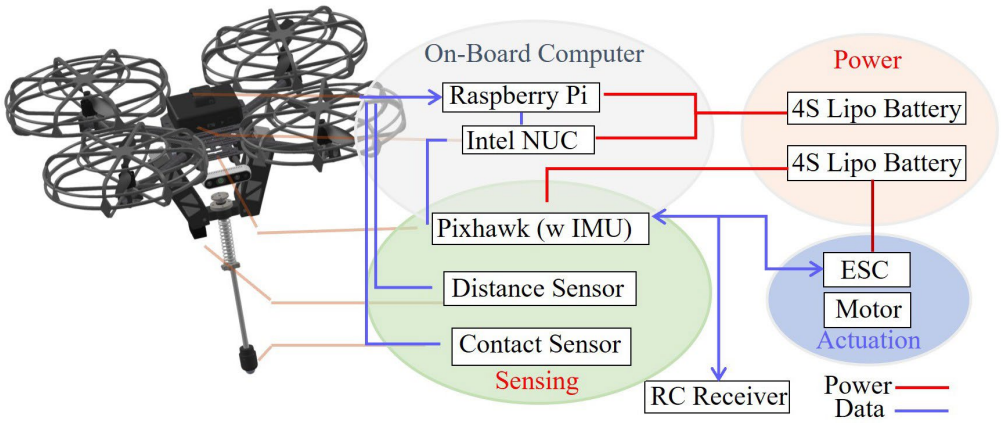

Figure 2.2: Hardware components of PogoX. (figure converted from Wang et al. (2024))

2.4.1 Flying Control of PogoX

In this section, we provide a short overview of the flying control framework of PogoX. During the flight phase, the dynamics of PogoX can be represented by a single rigid-body dynamcics (SRB) model with four actuators. State-of-the-art quadrotor control methods can be directly applied to PogoX.

Flight control is achieved in a cascade control manner by using an off-the-shelf, onboard PX4 flight controller operating in off-board control mode. PD controllers are employed for cascaded position and velocity control along three axes. The resulting desired orientation and total thrust are then sent to the PX4 for off-board orientation control. A detailed implementation of the PX4 control framework can be found in Meier et al. (2015).

2.4.2 Hopping Control of PogoX

In this section, we provide a short overview of the hopping control framework of PogoX as shown in Fig. 2.3. The hopping controller is decoupled into a vertical energy (or height) controller and a horizontal velocity controller.

The vertical energy or height controller is responsible for maintaining a desired apex height or tracking a given height trajectory. A PD controller can be employed for trajectory tracking, while a Control Lyapunov Function-Quadratic Program (CLF-QP) based energy controller can be used to asymptotically stabilize the apex height tracking error by tracking the corresponding energy target. In this thesis, since Data-Enabled Predictive Control (DeePC) based controllers require a data collection process for functioning, various height commands are given and instead of energy tracking, we leverage the PD height controller with handcrafted trajectories for data collection.

The horizontal velocity controller is designed to stabilize the hopping behavior by regulating the horizontal apex velocity. A step-to-step (S2S) dynamics-based stepping controller is employed. A nominal hopping gait is first optimized from the vertical energy-controlled Spring-Loaded Inverted Pendulum (SLIP) model, which provides a reference horizontal velocity for tracking. A standard deadbeat controller is then used to rapidly stabilize the error between the nominal and actual horizontal velocity, using the leg angle at the touchdown event as input. A continuous leg angle trajectory is then designed through interpolation using Bézier polynomials. The interpolated trajectory is divided into two segments: one interpolating from the liftoff leg angle to the apex leg angle, and the other from the apex leg angle to the desired touchdown angle. The leg angle is then tracked by the orientation controller provided by PX4.

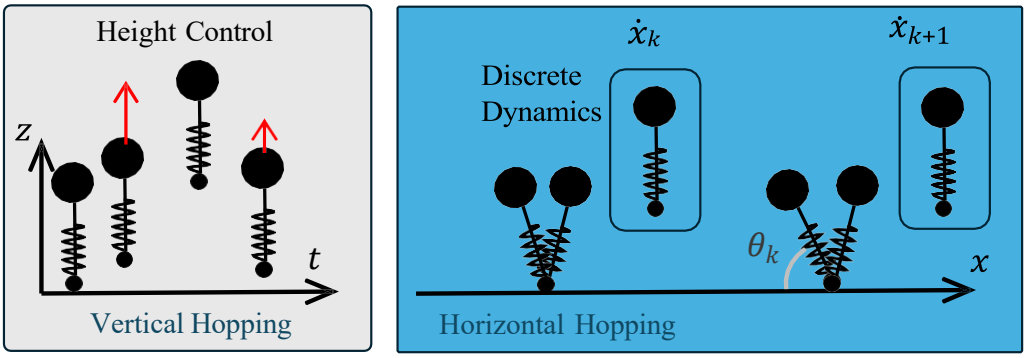

Figure 2.3: Hopping control framework of PogoX (figure converted from Wang et al. (2024)).

2.4.3 Overall Control Framework

The previously discussed flying and hopping control frameworks can be combined to synthesize the flying-hopping locomotion behaviors demonstrated in 2.4. The transition between different locomotion behaviors can be easily controlled by sending different RC signals. The more detailed description of the control framework of PogoX can be found in Wang et al. (2024) and Zeng et al. (2024).

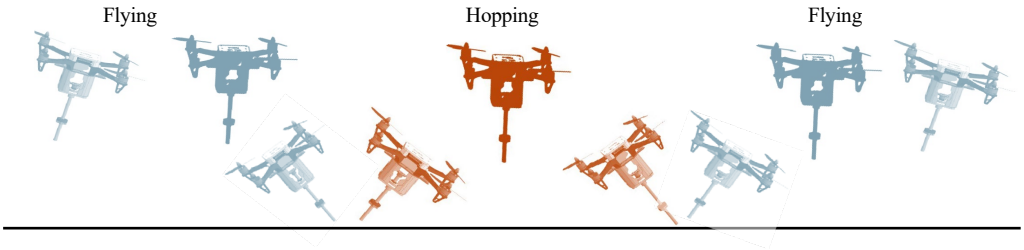

Figure 2.4: Overall behavior of flying-hopping locomotion (figure converted from Wang et al. (2024)).

3 data-augmented control of pogox hopping on dynamic terrains

In this chapter, we present Data-Augmented Control (DAC) framework that leverages DeePC to model robot-environment interaction dynamics and integrate them into the hybrid trajectory optimization and predictive control schemes. This approach allows for the generation of more dynamically feasible and easily trackable real-world trajectories, thereby improving control quality when interacting with external environments.

We begin by introducing the Data-Augmented Dynamics formulation for modeling vertical ground dynamics as a variation of the previously proposed slack PEM-MPC formulation. Then, we present the data-augmented trajectory optimization for 1D PogoX hopping on dynamic terrains such as in Sec. 3.2, followed by the online data-augmented predictive tracking controller in Sec. 3.4. Finally, simulation and hardware results of 1D PogoX hopping are provided in Sec. 3.5 to further demonstrate the effectiveness of the proposed framework. The Data-Augmented Control (DAC) pipeline is demonstrated in Fig. 3.2.

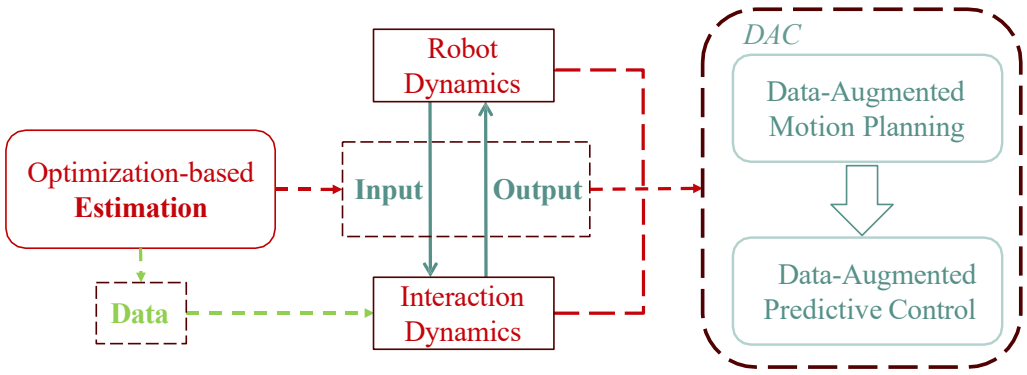

Figure 3.1: General Framework of DAC.

3.1 Data-Augmented Interaction Dynamics

Since explicitly describing the ground interaction dynamics in analytical form is challenging, we leverage Willems' fundamental lemma 2.5 to obtain a linear approximation of the vertical ground deformation dynamics. Approximating ground dynamics as linear is reasonable, as soft terrain dynamics is often modeled using compliant spring-damper models, as shown in Fahmi et al. (2020) and Neunert (2018), where an inherent linear property is assumed. However, spring-damper models are static and instantaneous, which means they lack state transitions between different time steps. This leads to highly discontinuous forces at each time step, which contradicts the observed smooth deformation behavior of soft terrain.

Therefore, in this thesis, we model the ground dynamics from data and conduct comparisons between the proposed method and conventional spring-damper modeling in both simulation and hardware experiments to demonstrate the superiority of the proposed approach. Inspired by the spring-damper models for ground deformation, we choose the system inputs as ground deformation d and its rate of change $\dot{d}$. The ground reaction force generated by the dynamics is denoted as F_g. Following the same procedure as demonstrated in Sec. 2.3, the ground deformation dynamics can be expressed as:

$$\begin{pmatrix} g \\ \sigma_y \end{pmatrix} = G^{-1}A^T(AG^{-1}A^T)^{-1}b, \tag{3.1}$$

where G, A are the matrices with the same meanings defined in the previous section, while $b = \begin{bmatrix} d_{ini} & \dot{d}_{ini} & F_{g_{ini}}^T & d_{[0:T]} & \dot{d}_{[0:T]} & F_{g_{[0:T]}}^T \end{bmatrix}^T$, and the subscript $[0:T]$ denotes a sequence of data with length T. Since in this formulation we are not concerned with the value of σ_y, we can let

$B = \begin{bmatrix} I \\ Y_f \end{bmatrix} \quad 0$ and redefine the data transition matrix as:

$$G_{slack} = BG^{-1}A^T(AG^{-1}A^T)^{-1}. \tag{3.2}$$

Then the ground deformation dynamics can be written as:

$$^{g}F_{[0:T]} = G_{slack} \begin{bmatrix} d_{ini} \\ \dot{d}_{ini} \\ F^{g}_{ini} \\ d_{[0:T]} \\ \dot{d}_{[0:T]} \end{bmatrix}. \tag{3.3}$$

However, if the ground deformation dynamics is formulated in this manner, accurately estimating both the ground deformation and the corresponding ground reaction force becomes extremely challenging. To sim-

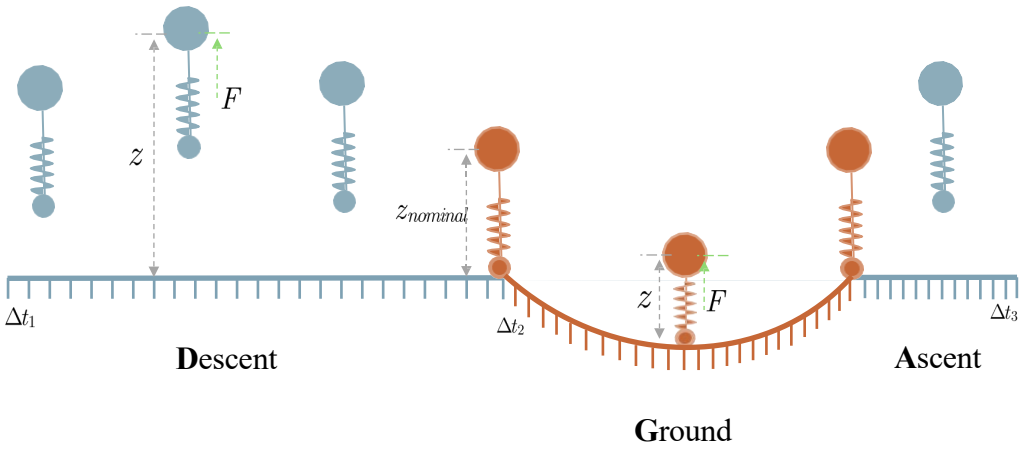

Figure 3.2: Demonstration of input (gray) output (green) of data augmented dynamics in flight phase (blue) and ground phase (red).

plify state estimation, we adopt a data-driven approach to model the ground deformation and the spring-damper system as a unified entity. Leveraging the concept that a series of springs can be represented as an

equivalent single spring, we assume that the combined dynamics of the spring-damper system and the ground remains linear. As depicted in Fig. 3.2, the input of the augmented dynamics is now defined as the relative position between the ground surface and the center of mass (COM) height of the robot and its rate of change. The output is defined as the force exerted on the COM of the robot. The data augmented dynamics can now be written as:

$$F_{[0:T]} = G_{slack} \begin{bmatrix} z_{ini} - z_{nominal} \\ \dot{z}_{ini} \\ F_{ini} \\ z_{[0:T]} - z_{nominal} \\ \dot{z}_{[0:T]} \end{bmatrix}, \tag{3.4}$$

where $z_{nominal} \in R^{T_{ini}}$.

3.2 Data-Augmented Motion Planning

We present the offline trajectory optimization formulation for one-dimensional PogoX hopping on an air mattress. This module plans an optimal motion that enables highly dynamic jumping by accounting not only for the robot's dynamics but also for the interaction dynamics with the environment.

3.2.1 Hybrid Trajectory Optimization Formulation

Since hopping locomotion is naturally governed by hybrid dynamics behaviors, the trajectory optimization problem formulated in this section is a Hybrid Trajectory Optimization (HTO) problem. We adopt the same approach as described in Schultz and Mombaur (2010) and Posa et al. (2016), where a fixed mode sequence is prescribed, and discrete state transitions are incorporated at the end of each mode.

The mode sequence adopted in this thesis for hopping consists of a descent phase with a touchdown event as the guard condition, followed

by a ground phase with a liftoff event as the guard condition, and finally an ascent phase, which is constrained to return to the same state where the descent phase begins. This setup allows the hopping behavior to be generalized to multi-cycle hopping by simply repeating the optimized trajectory arbitrary number of times for tracking.

To further enhance the energy efficiency of the trajectory optimization problem, we incorporate the length of time steps of the descent and ascent phases as optimization variables, allowing the determination of optimal descent and ascent durations. Additionally, we impose minimum and maximum time constraints to guide the search:

$$\begin{aligned} T &= N_1\Delta t_1 + N_2\Delta t_2 + N_3\Delta t_3, \\ T_{min} &\leqslant T \leqslant T_{max}. \end{aligned} \tag{3.5}$$

where N_1, N_2 and N_3 represents the number of collocation points for descent, ground and ascent phase, respectively, while Δt_1, Δt_2 and Δt_3 denote the corresponding time steps for each phases.

Since Willems' fundamental lemma is formulated for discrete-time LTI systems, the time step size of the predicted input-output trajectories must match the time step used when collecting input-output data. This prevents us from directly optimizing Δt_2 in the trajectory optimization formulation. To mitigate this issue, we propose to optimization number of grid points rather than the time step size. A non-smooth, nonlinear optimization problem can be formulated where its goal is to optimize over the optimized cost function with respect to N_2:

$$\min_{N_2} \quad J(N_2), \tag{3.6}$$

where the detailed formulation of J can be found below. This optimization problem is solved by Bayesian optimization where detailed descriptions can be found in Frazier (2018) and Mockus (2012). It's a gradient-free

method that can solve for the global optimal of an optimization problem.

Descent Phase

The dynamics of descent phase of hopping is described by an actuated single-rigid-body model:

$$m\ddot{z} = u - mg, \tag{3.7}$$

where m is the mass, $\ddot{z}$ is the z direction acceleration, u is the total thrust force and g = 9.81 is the gravitational acceleration. The spring deformation, ground deformation, and their rates of change are constrained to zero, effectively eliminating the ground reaction force. The initial condition of the descent phase is constrained to a the desired apex height with zero linear velocity. For the guard condition that provides a smooth transition between the descent phase and ground phase, we constraint the height of the robot COM to be the height of PogoX when the foot is in contact with the ground but the spring is still uncompressed. The constraint is written as: $z_{N_1} = L_{max}$ where L_{max} is the length of the robot when spring is uncompressed.

Ground Phase

The dynamics of the ground phase of hopping are described by a 1D-SLIP model, where the spring is assumed to be massless, and the ground reaction force is equal in magnitude to the spring force. As previously mentioned, a DAC based approach is used to model the combined effects of all forces acting on the robot's center of mass (COM). Consequently, the system is represented as an actuated SRB model with external forces:

$$m\ddot{z} = F + u - mg. \tag{3.8}$$

Ascent Phase

The dynamics of ascent phase of hopping is the same to the descent phase where boundary constraints are added to make sure that PogoX hops back to the desired apex height with zero velocity. It is noted that the total deformation is allowed to freely move in this phase but the spring force is always constrained to zero.

Collocation

Since the dynamics of the trajectory optimization problem is written in a continuous form, we need to transform it into a nonlinear program (NLP) for easy solution of standard optimization solvers. In this thesis, we choose trapezoidal collocation method for its computational efficiency and easy-to-implement feature. The collocation rule is written as below.

$$\begin{aligned} z_{k+1} &= z_k + \frac{\Delta t}{2}(\dot{z}_{k+1} + \dot{z}_k), \\ \dot{z}_{k+1} &= \dot{z}_k + \frac{\Delta t}{2}(\ddot{z}_{k+1} + \ddot{z}_k), \end{aligned} \tag{3.9}$$

where $k \in (0, N_1 + N_2 + N_3 - 1)$ and Δt represents the corresponding time step size.

Control Bound

To ensure the PX4 system remains functional for the optimized trajectories, we impose a constraint that keeps the total thrust force within a specified range, maintaining it above a minimum threshold while staying below a maximum limit.

Objective Function

The objective function of this optimization problem is set to purely minimize the control efforts with respect to the optimization variables $y =$

$\{\Delta t_1, \Delta t_3, z, \dot{z}, F, u\}$ which is written as:

$$J(N_2) = \min_{y} \sum_{i=0}^{N_1+N_2+N_3} ||u_i||^2. \tag{3.10}$$

3.3 Optimization-Based State Estimation

In this section, we present the formulation of the state estimator used to provide both the robot states and the input-output data required for modeling interaction dynamics. In this thesis, we adopt an optimization-based estimation method known as Moving Horizon Estimation (MHE). MHE enables simultaneous real-time estimation of both the external force exerted on the robot's base and the robot's floating-base states, thereby facilitating the online implementation of the DAC framework.

State of the robot is defined as: $\mathbf{x} = \begin{bmatrix} x^T & b_a^T & m^T & F^T \end{bmatrix}^T$, where x represents the robot states, including position and velocity of the floating base in the world frame, b_a is the IMU accelerometer bias in the body frame, m is the generalized momentum, and F denotes the ground reaction force (GRF), also expressed in the world frame. Since MHE is formulated as a windowed optimization problem, a slding window of N states are given as follow: $\mathbf{x}_{[T-N,T]} = \begin{bmatrix} \mathbf{x}_{T-N} & \mathbf{x}_{T-N+1} & \dots & \mathbf{x}_N \end{bmatrix}$. Following the setup in Kang et al. (2024), the MHE problem is formulated in a decentralized manner, where the robot's orientation is obtained either through a quaternion-based Extended Kalman Filter (QEKF) or from a motion capture system. Given that orientation is treated as known at each time step within the sliding window, the momentum and IMU dynamics are formulated as time-varying linear constraints. The resulting decentralized

MHE formulation is expressed as:

$$\min_{\mathbf{x}_{[T-N,T]}} \Gamma(\mathbf{x}_{T-N}) + \sum_{k=T-N}^{T-1} \|\delta_k^x\|_{Q_k} + \sum_{k=T-N}^{T} \|\delta_k^y\|_{R_k}, \qquad \text{(MHE)}$$

$$\text{s.t.} \quad F_k \geqslant 0, \forall k \in \{T-N, ..., T\},$$

$$y_k = \mathbf{H}\mathbf{x}_k + \delta_k^y, \ \forall k \in \{T-N, ..., T\},$$

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k u_k + \delta^x{}_k \ \forall k \in \{T-N, ..., T-1\}.$$

where Γ denotes the arrival cost, which plays a critical role in ensuring the stability of the estimator and is generally computed using an Extended Kalman Filter (EKF). $\mathbf{A}_k$, $\mathbf{B}_k$ represent the time-varying linearizations of the momentum and IMU dynamics with respect to the robot's orientation. $\mathbf{H}_k$ is the measurement matrix associated with the motion capture system, which also provides direct measurements of the robot's pose and velocity. During the contact phase, the ground reaction forces are constrained to be non-negative, as no dragging forces can be generated. Further details on each component can be found in our previous work [37].

3.4 Online Data-Augmented Predictive Control (DAPC)

In this section, we demonstrate the formulation of data-augmented model predictive controller for online tracking the trajectories generated by the Data-Augmented Motion Planning module.

3.4.1 Objective Function

Unlike the previously presented trajectory optimization formulation, the objective function of DAPC is defined as:

$$\min_{y} \|x - x^{ref}\|_Q^2 + \|u\|_R^2, \qquad (3.11)$$

where $x = \{z, \dot{z}\}$, $y = \{x, F, u, z_g, \dot{z}_g, F_g\}$ are optimization variables.

3.4.2 Contact Sequence

Similar to trajectory optimization, the formulation of DAPC is a model predictive control (MPC) problem for hybrid dynamical systems, as described in Lazar (2006), Camacho et al. (2010). In this framework, the system dynamics applied at each time step can vary according to a predefined contact sequence. When the robot is in contact with the ground, it is considered to be in the ground phase, and all optimization variables correspond to this phase are denoted with the left superscript g. Conversely, when the robot is not in contact with the ground, it is in the flight phase, and all optimization variables correspond to this phase are denoted with the left superscript f. By assuming that the window length of DAPC is shorter than the optimized ground phase time, four types of contact sequences will present for DAPC formulation:

- Ground phase
- Ground phase transit to flight phase
- Flight phase
- Flight phase transit to ground phase

3.4.3 Dynamics Constraint

The dynamics constraint in DAPC is identical to the trajectory optimization formulation previously described for the ground phase. The same forward Euler method is used to discretize the continuous-time second-order dynamics, applying them at each time step. The data-augmented dynamics constraint introduced in Section 3.1 is formulated as an external dynamics constraint.

However, the initial trajectories used in the data-augmented dynamics are not always initialized to zero. If the prescribed contact sequence begins with the flight phase, the initial trajectories are set to zero, corresponding to a zero initial condition. In contrast, if the contact sequence starts with the ground phase, the initial trajectories are initialized using the actual center of mass (COM) height, velocity, and external force history.

To ensure consistency with the hybrid nature of the dynamics and the data-augmented model, additional constraints are incorporated based on the prescribed contact sequence to regulate the behavior of external forces accordingly.

Flight Phase

Constraints on external force are imposed to ensure that no external forces act on the robot. The constraints are written as:

$$^{f}F = 0. \tag{3.12}$$

Additional constraints are imposed on $^{f}z_g$ and $^{f}\dot{z}_g$ to ensure that, during transitions from the flight phase to the ground phase, the corresponding ground reaction force $^{f}F_g$ remains zero. This guarantees that the subsequently optimized $^{c}F_g$ is physically consistent. The constraints are formulated as follows:

$$\begin{aligned} ^{f}z_g &= 0, \\ ^{f}\dot{z}_g &= 0. \end{aligned} \tag{3.13}$$

Ground Phase

Constraints on the external force are imposed to ensure consistency with the data-augmented dynamics. Simultaneously, the vertical position z and velocity $\dot{z}$ are constrained to match the auxiliary variables z_g and $\dot{z}_g$ which are the inputs to the augmented dynamics constraint. These

constraints are formulated as follows:

$$\begin{aligned} {}^{c}F_g &= {}^{c}F, \\ {}^{c}z_g &= {}^{c}z, \\ {}^{c}\dot{z}_g &= {}^{c}\dot{z}. \end{aligned} \tag{3.14}$$

If the contact sequence involves a transition from contact to flight, the constraints ensure that the auxiliary variables ${}^{f}F_g$ are naturally rolled out during the flight phase by setting the inputs ${}^{f}z_g$ and ${}^{f}\dot{z}_g$ to zero. This guarantees that the flight phase constraints do not interfere with the prediction of ground-phase external forces under the augmented dynamics constraint.

3.4.4 Initial Condition

Initial condition constraints are imposed on the center of mass (COM) height and velocity, z and $\dot{z}$, to ensure that the initial states match the actual values:

$$\begin{aligned} z(0) &= z_0, \\ \dot{z}(0) &= \dot{z}_0. \end{aligned} \tag{3.15}$$

3.4.5 Control Input

Similar to the trajectory optimization formulation, we enforce minimum and maximum total thrust bounds to make sure the control inputs is realizable on real hardware:

$$u_{min} \leqslant u \leqslant u_{max}. \tag{3.16}$$

Since the entire optimization problem is formulated along a single direction of motion getting rid of the complexity of orientation representation, with dynamics modeled as a linear time-invariant (LTI) system

within each contact phases, and all equality and inequality constraints are linear, the resulting problem is a quadratic program (QP). This structure allows the optimization to be efficiently solved online using standard QP solvers.

3.5 Results

The effectiveness of the proposed framework has been validated in both simulation and hardware experiments.

The trajectory optimization formulation is implemented in MATLAB using the CasADi Andersson et al. (2019) toolbox, with IPOPT Wächter et al. (2002) selected as the optimization solver. The Bayesian optimization problem is solved using MATLAB's Bayesian Optimization Toolbox. The DAPC framework is implemented in Python, also using CasADi, with OSQP Stellato et al. (2020a) chosen as the solver and the problem can be solved for approximately 150Hz online. The optimized trajectories generated by the trajectory optimization process are saved as readable log files. During execution, the DAPC script performs linear interpolation on logged trajectories based on the current time within the trajectory cycle to obtain reference values. Event-based contact detection is employed to ensure that the correct desired trajectories and corresponding contact sequences are sent to DAPC for accurate tracking.

ROS 2 described in Macenski et al. (2022) is used as the communication middleware, with a custom-built interface developed for integration with the MuJoCo simulator described in Todorov et al. (2012). The low-level controller is custom-implemented in simulation and provided by PX4 on the hardware platform.

Simulation

The air mattress is simulated in MuJoCo using its deformable object simulation functions. The robot is restricted to a z-directional motion by assuming a perfect orientation control. Input-output trajectories for constructing the data transition matrix are collected from controlling PogoX to jump on the air mattress with different desired tracking heights. Totally 50 cycles of jumping data are collected with the desired tracking heights ranging from 1.0m to 1.6m.

Fig.3.3 has depicted the prediction capacity of the proposed method by plotting the predicted output trajectories alongside the corresponding actual output trajectories. The predicted trajectories are obtained by directly rolling out the augmented dynamics over a horizon of 10, with the initial trajectory length set to 30. The data transition matrix is constructed from data collected at heights ranging from 1.0m to 1.5m, while the test data comes from a dataset collected at a tracking height of 1.6m.

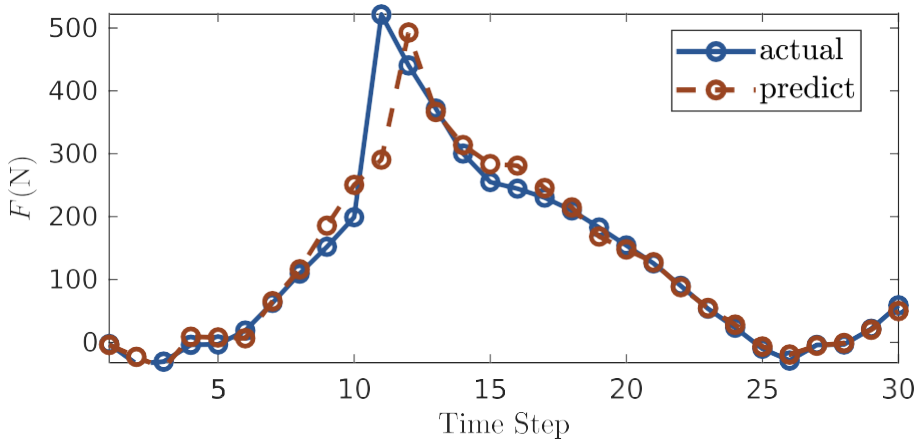

Figure 3.3: Plot of predicted trajectories (red) versus actual trajectories (blue).

Fig. 3.4 illustrates the trajectory tracking performance of base height, velocity, and contact sequence, highlighting the effectiveness of the pro-

posed framework in actively leveraging augmented dynamics for improved trajectory tracking. The prediction horizon in the DAPC formulation is set to 20, while the initial trajectory length for the augmented dynamics constraint is set to 30. The data transition matrix is constructed from a diverse set of jumping motions, comprising a total of 1000 data points collected at a frequency of 200Hz. The weights of objective function are configured as $Q = \text{diag}(5e6, 5e5)$, $R = 5e1$.

Hardware

In the hardware experiments, we validate the effectiveness of the proposed method by testing PogoX performing jumping maneuvers on an air mattress. The experimental setup is illustrated in Fig. 3.5(a), where the robot is commanded to jump vertically onto the air mattress. The horizontal position and orientation are stabilized using the previously described Step-to-Step (S2S) controller. For data collection, a PD height controller is used to track various height setpoints during jumping. The controller gains and execution frequency are kept consistent with the simulation setup to ensure a fair comparison.

Figure 3.5(b) shows the trajectory tracking results, where the reference trajectory is generated using the Data-Augmented Motion Planning (DAMP) module. The length of the initial prediction horizon is also kept identical to that in the simulation experiments. The results indicate that both position and velocity are well-tracked during the flight phase. However, slight tracking mismatches are observed around the liftoff events, which are believed to arise from discrepancies between the idealized spring-damper model used in the control formulation and the real interaction dynamics experienced by the robot.

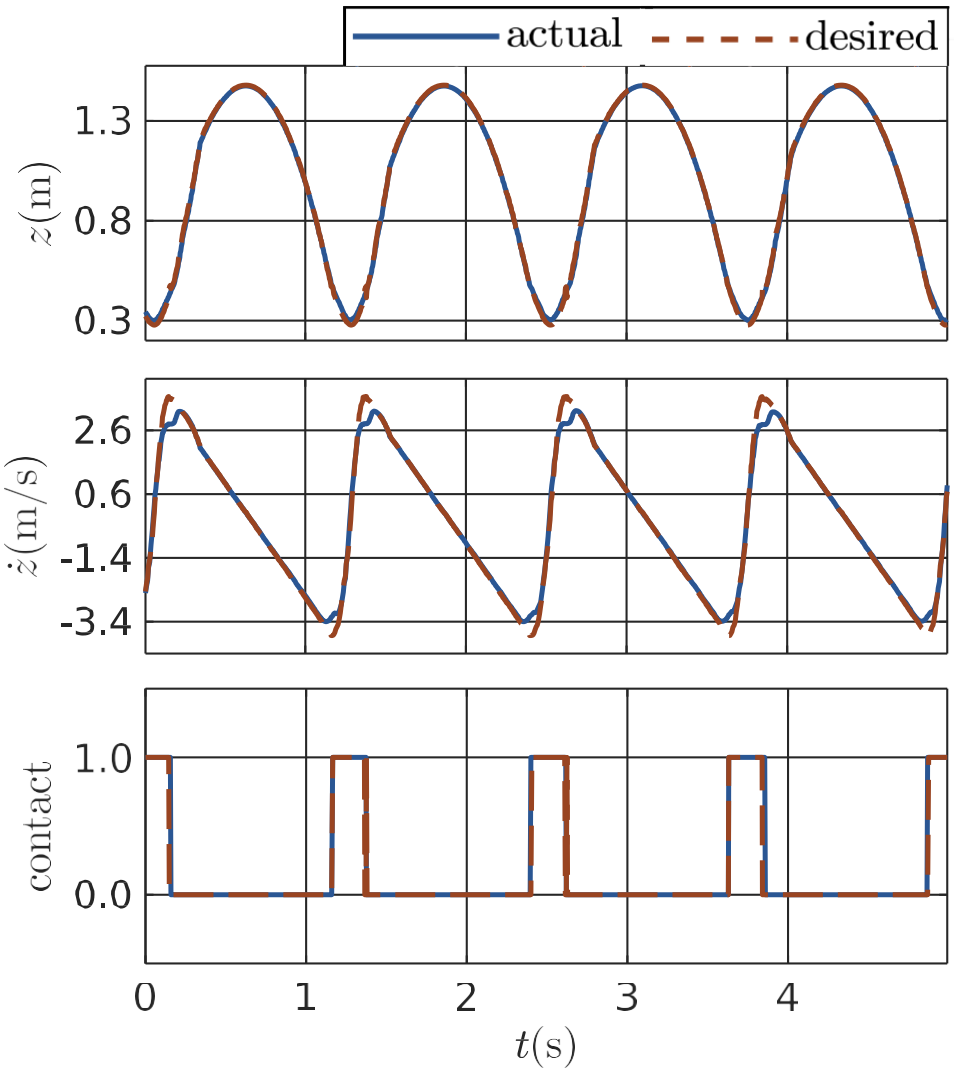

Figure 3.4: Plot of trajectory tracking results in simulation.

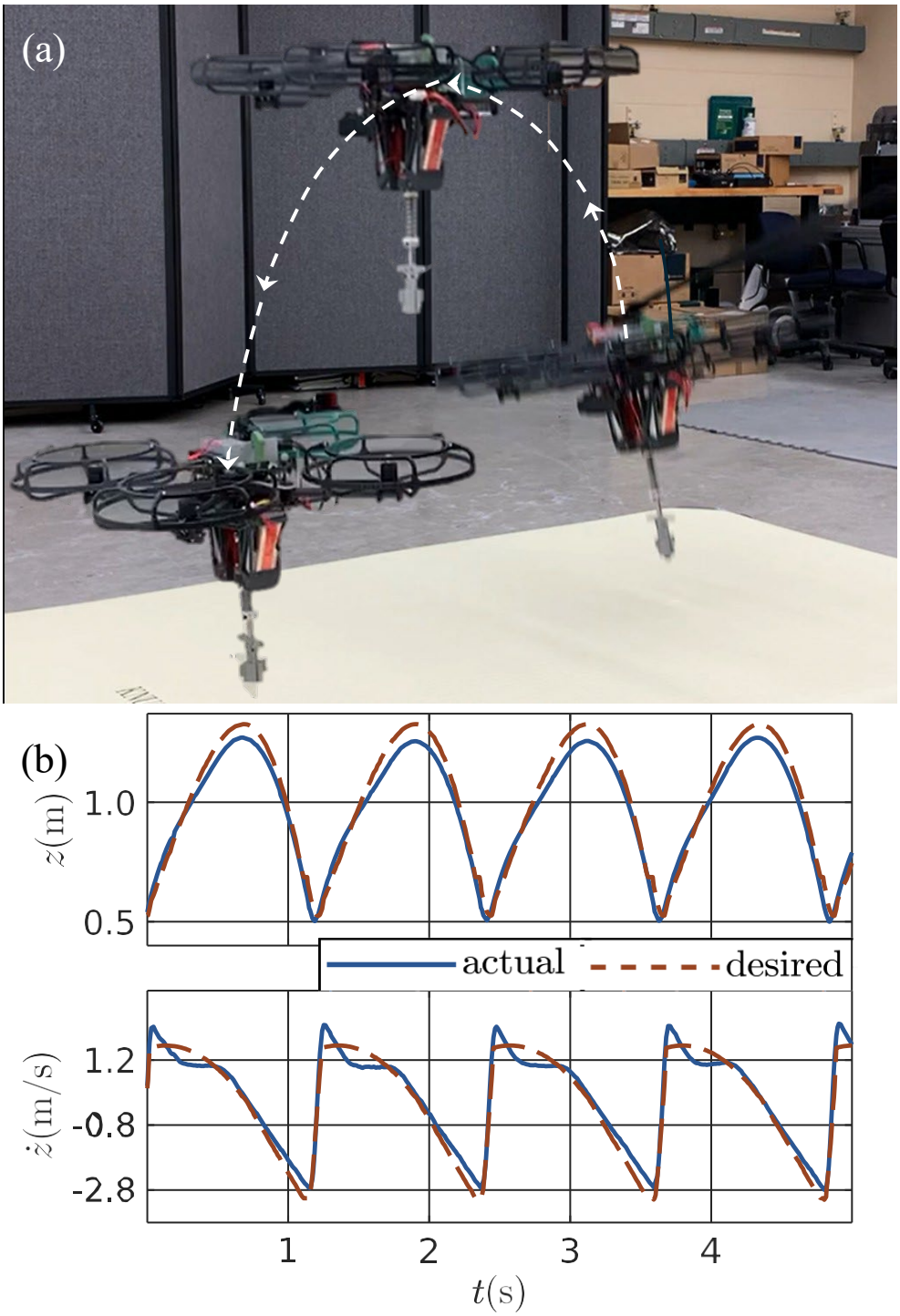

Figure 3.5: Results of hardware air mattress hopping: (a) demonstrates the setups of hardware experiment. (b) is the plot of trajectory tracking results on hardware.

4 summary

In this thesis, we presented a general Data-Augmented Control (DAC) framework designed to mitigate the performance limitations of model-based controllers by actively incorporating data-driven interaction dynamics into both the motion planning and predictive control modules. To further enhance performance, Bayesian optimization was employed to determine the optimal ground-phase duration, enabling more effective exploitation of the augmented dynamics. The proposed methods were supported by in-depth mathematical analysis for qualitative validation, and their effectiveness was quantitatively demonstrated through extensive simulation and hardware experiments involving dynamic jumping of the robot on an air mattress.

4.0.1 Future Work

Future research will focus on generalizing the proposed frameworks to a broader class of robotic systems, particularly those with highly nonlinear dynamics.

To handle time-varying dynamics, it is essential to develop techniques for online updating of Hankel matrices and the corresponding data transition matrices, ensuring smooth transitions in implicitly identified models. Moreover, certifying the quality of online-collected data and analyzing the associated robustness and adaptivity will be crucial for real-world deployment.

references

Alsalti, Mohammad, Victor G. Lopez, and Matthias A. Müller. 2023. On the design of persistently exciting inputs for data-driven control of linear and nonlinear systems. *IEEE Control Systems Letters* 7:2629–2634.

Andersson, Joel A E, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. 2019. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation* 11(1):1–36.

Bertec. 2020. Bertec: https://www.bertec.com/.

Brescianini, Markus; D'Andrea Raffaello, Dario; Hehn. 2013. Nonlinear quadrocopter attitude control technical report.

Camacho, E.F., D.R. Ramirez, D. Limon, D. Muñoz de la Peña, and T. Alamo. 2010. Model predictive control techniques for hybrid systems. *Annual Reviews in Control* 34(1):21–31.

Carron, Andrea, Elena Arcari, Martin Wermelinger, Lukas Hewing, Marco Hutter, and Melanie N. Zeilinger. 2019. Data-driven model predictive control for trajectory tracking with a robotic arm. *IEEE Robotics and Automation Letters* 4(4):3758–3765.

Coulson, Jeremy, John Lygeros, and Florian Dörfler. 2018. Data-enabled predictive control: In the shallows of the deepc. *2019 18th European Control Conference (ECC)* 307–312.

Di Carlo, Jared, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. 2018. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 ieee/rsj international conference on intelligent robots and systems (iros)*, 1–9.

Didier, Alexandre, Anilkumar Parsi, Jeremy Coulson, and Roy S. Smith. 2021. Robust adaptive model predictive control of quadrotors. In *2021 european control conference (ecc)*, 657–662.

Elokda, Ezzat, Jeremy Coulson, Paul Beuchat, John Lygeros, and Florian Dörfler. 2021. Data-enabled predictive control for quadcopters. *International Journal of Robust and Nonlinear Control* 31.

Fahmi, Shamel, Michele Focchi, Andreea Radulescu, Geoff Fink, Victor Barasuol, and Claudio Semini. 2020. Stance: Locomotion adaptation over soft terrain. *IEEE Transactions on Robotics* 36(2):443–457.

Fawcett, Randall, Kereshmeh Afsari, Aaron Ames, and Kaveh Akbari Hamed. 2022. Toward a data-driven template model for quadrupedal locomotion. *IEEE Robotics and Automation Letters* 7:1–8.

Frazier, Peter I. 2018. A tutorial on bayesian optimization. 1807.02811.

Huang, Linbin, Jeremy Coulson, John Lygeros, and Florian Dörfler. 2019. Data-enabled predictive control for grid-connected power converters. In *2019 ieee 58th conference on decision and control (cdc)*, 8130–8135.

Huang, Linbin, Jianzhe Zhen, John Lygeros, and Florian Dörfler. 2023. Robust data-enabled predictive control: Tractable formulations and performance guarantees. *IEEE Transactions on Automatic Control* 68(5):3163–3170.

Kang, Jiarong, Yi Wang, and Xiaobin Xiong. 2024. Fast decentralized state estimation for legged robot locomotion via ekf and mhe. *IEEE Robotics and Automation Letters* 9(12):10914–10921.

Khadiv, Majid, S. Ali A. Moosavian, Aghil Yousefi-Koma, Majid Sadedel, Abbas Ehsani-Seresht, and Saeed Mansouri. 2019. Rigid vs compliant contact: an experimental study on biped walking. *Multibody System Dynamics* 45(4):379–401.

Khalil, Wisama, and Fouad Bennis. 1995. Symbolic calculation of the base inertial parameters of closed-loop robots. *The International Journal of Robotics Research* 14(2):112–128. https://doi.org/10.1177/027836499501400202.

Khazoom, Charles, Seungwoo Hong, Matthew Chignoli, Elijah Stanger-Jones, and Sangbae Kim. 2024. Tailoring solution accuracy for fast whole-body model predictive control of legged robots. *IEEE Robotics and Automation Letters* PP:1–8.

Lazar, M. 2006. Model predictive control of hybrid systems : stability and robustness. Phd thesis 1 (research tu/e / graduation tu/e), Electrical Engineering.

Le Lidec, Quentin, Wilson Jallet, Louis Montaut, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. 2024. Contact models in robotics: A comparative analysis. *IEEE Transactions on Robotics* 40:3716–3733.

Leboutet, Quentin, Julien Roux, Alexandre Janot, Julio Rogelio Guadarrama-Olvera, and Gordon Cheng. 2021. Inertial parameter identification in robotics: A survey. *Applied Sciences* 11(9).

Lynch, Kevin M., and Frank C. Park. 2017. *Modern robotics: Mechanics, planning, and control*. 1st ed. USA: Cambridge University Press.

Macenski, Steven, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. 2022. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics* 7(66):eabm6074.

Marhefka, D.W., and D.E. Orin. 1999. A compliant contact model with nonlinear damping for simulation of robotic systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 29(6):566–572.

Meier, Lorenz, Dominik Honegger, and Marc Pollefeys. 2015. Px4: A node-based multithreaded open source robotics framework for deeply

embedded platforms. In *2015 ieee international conference on robotics and automation (icra)*, 6235–6240.

Mellinger, Daniel, and Vijay Kumar. 2011. Minimum snap trajectory generation and control for quadrotors. In *2011 ieee international conference on robotics and automation*, 2520–2525.

Mockus, J. 2012. *Bayesian approach to global optimization: Theory and applications*. Mathematics and its Applications, Springer Netherlands.

Murray, Richard M., S. Shankar Sastry, and Li Zexiang. 1994. *A mathematical introduction to robotic manipulation*. 1st ed. USA: CRC Press, Inc.

Neunert, Markus Giftthaler Markus Bellicoso Carmine D. Carius Jan Gehring Christian Hutter Marco Buchli Jonas, Michael Stäuble. 2018. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robotics and Automation Letters* 3(3):1458–1465.

Neunert, Michael, Markus Stäuble, Markus Giftthaler, Carmine D. Bellicoso, Jan Carius, Christian Gehring, Marco Hutter, and Jonas Buchli. 2018. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robotics and Automation Letters* 3(3):1458–1465.

Optitrack. 2020. Optitrack: https://optitrack.com/.

Posa, Michael, Scott Kuindersma, and Russ Tedrake. 2016. Optimization and stabilization of trajectories for constrained dynamical systems. In *2016 ieee international conference on robotics and automation (icra)*, 1366–1373.

Russ, Tedrake. 2023. *Underactuated robotics*.

———. 2024. *Robotic manipulation*.

Schultz, Gerrit, and Katja D. Mombaur. 2010. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics* 15:783–792.

Simpkins, Alex. 2012. System identification: Theory for the user, 2nd edition (ljung, l.; 1999) [on the shelf]. *IEEE Robotics Automation Magazine* 19(2):95–96.

Sreenath, Koushil, Taeyoung Lee, and Vijay Kumar. 2013. Geometric control and differential flatness of a quadrotor uav with a cable-suspended load. In *52nd ieee conference on decision and control*, 2269–2274.

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

Stellato, Bartolomeo, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. 2020b. Osqp: An operator splitting solver for quadratic programs. *Mathematical Programming Computation* 12(4):637–672.

Sun, Sihao, Angel Romero, Philipp Foehn, Elia Kaufmann, and Davide Scaramuzza. 2022. A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight. *IEEE Transactions on Robotics* 38(6):3357–3373.

Todorov, Emanuel, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 ieee/rsj international conference on intelligent robots and systems*, 5026–5033.

Wang, Yi, Jiarong Kang, Zhiheng Chen, and Xiaobin Xiong. 2024. Terrestrial locomotion of pogox: From hardware design to energy shaping and step-to-step dynamics based control. In *2024 ieee international conference on robotics and automation (icra)*, 3419–3425.

Wegner, Felix. 2021. Data-enabled Predictive Control of Robotic Systems. Master's thesis, ETH Zurich.

Wensing, Patrick M., Günter Niemeyer, and Jean-Jacques E. Slotine. 2024. A geometric characterization of observability in inertial parameter identification. *The International Journal of Robotics Research* 43(14):2274–2302. https://doi.org/10.1177/02783649241258215.

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

Wächter, A, Lorenz Biegler, Yi-dong Lang, and Arvind Raghunathan. 2002. Ipopt: An interior point algorithm for large-scale nonlinear optimization.

Xie, Zhongqu, Long Li, and Xiang Luo. 2022. A foot-ground interaction model based on contact stability optimization for legged robot. *Journal of Mechanical Science and Technology* 36:921–932.

Zeng, Yicheng, Yuhao Huang, and Xiaobin Xiong. 2024. Reference-steering via data-driven predictive control for hyper-accurate robotic flying-hopping locomotion. 2411.18793.