Robust Agility via Learned Zero Dynamics Policies

Noel Csomay-Shanklin[1*], William D. Compton[1*], Ivan Dario Jimenez Rodriguez[1*],
Eric R. Ambrose[2], Yisong Yue[1], Aaron D. Ames[1]

***Abstract*— We study the design of robust and agile controllers for hybrid underactuated systems. Our approach breaks down the task of creating a stabilizing controller into: 1) learning a mapping that is invariant under optimal control, and 2) driving the actuated coordinates to the output of that mapping. This approach, termed Zero Dynamics Policies, exploits the structure of underactuation by restricting the inputs of the target mapping to the subset of degrees of freedom that cannot be directly actuated, thereby achieving significant dimension reduction. Furthermore, we retain the stability and constraint satisfaction of optimal control while reducing the online computational overhead. We prove that controllers of this type stabilize hybrid underactuated systems and experimentally validate our approach on the 3D hopping platform, ARCHER. Over the course of 3000 hops the proposed framework demonstrates robust agility, maintaining stable hopping while rejecting disturbances on rough terrain.**

I. Introduction

The underactuated dynamics inherent to legged locomotion, swimming, and dexterous manipulation impose fundamental limits on controller performance and necessitate a critical understanding of the system's flow to achieve complex behaviors. Underactuation prevents arbitrarily shaping a system's dynamics, undermining the assumptions of many control-theoretic methods such as feedback linearization [1] and offline trajectory tracking. This work leverages recent advances in controller design for underactuated systems [2], [3], optimal control [4], and their integration with computational learning methods to design feedback strategies that exploit the structure of underactuation, enabling the agile and robust behavior shown in Figure 1.

A predominant method for controlling underactuated systems is Model Predictive Control (MPC) [5], [6], which leverages concepts from optimal control over a prediction horizon to achieve stabilization [7]. Performance of MPC controllers improves with longer horizons and finer time discretizations, both of which conflict with its strict real-time computational requirements. To address the high computational cost of full-model optimization problems, some methods leverage a gradation of model fidelities along a time horizon [8], [9]. Other methods rely on offline trajectory optimization to generate desirable behaviors, and then track these behaviors online [10]. For underactuated systems, the online tracking problem can be non-trivial, often requiring additional feedback mechanisms to stabilize the underactuated states such as regulators [11].

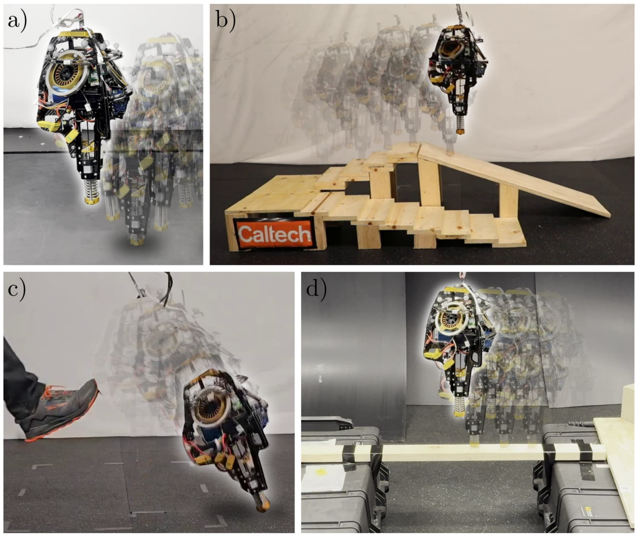

Fig. 1: Experiments run with Zero Dynamics Policies: a) treadmill hopping with disturbances up to 1 mile per hour, b) 1.5" stair climbing and 20° ramp descending, c) disturbance rejection, and d) hopping across a 2x4.

Reinforcement learning (RL) [12] takes the concept of offline computation even further, using concepts from stochastic optimal control and parallelized simulation environments to synthesize feedback controllers. RL methods have shown robust performance [13], [14] when the policy is trained in sufficiently randomized domains. Current methods in RL improve policies through simulator rollouts [15], typically at the expense of high data complexity. Although these can work well, they exhibit extreme sensitivity to cost function parameters and ignore the underlying system structure.

Heuristics, on the other hand, are able to leverage intuition about system structure, and can achieve stabilization with minimal online or offline computational overhead. In the context of legged locomotion, the Raibert Heuristic for hopping [16], inverted pendulum models for walking [17], and spring-loaded pendulums for running [18] all reason about where a legged robot's feet should be placed in order to stabilize the center of mass. While these methods may be less formal than the methods above and require significant domain expertise to implement, they tend to reason (perhaps implicitly) about the fundamental control structure needed to address the underactuation.

The above methods generally intersect in two places: first, an application of feedback to the actuated states based on the position of underactuated states (either explicitly or through replanning), and second, a dependence on optimality

*denotes equal contribution. [1]Authors are with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125. [2]Authors are with NASA Jet Propulsion Laboratory.

This research was supported by Technology Innovation Institute (TII), NSF Graduate Research Fellowship No. DGE-1745301, AeroVironment, NSF Grant No. 1918655 and Raytheon, Beyond Limits, JPL RTD 1643049.

to generate stable, desirable behaviors. We propose a method which combines these two ideas, using optimality to ensure stability while reasoning explicitly about the structure of underactuation. Specifically, we leverage the notion of *zero dynamics* to explicitly decompose the system into actuated and unactuated coordinates [19], [20], [21], [22]. We pair this paradigm with optimal control to learn a mapping from the unactuated state to a desired actuated state, termed a Zero Dynamics Policy (ZDP), which is then stabilized using a tracking controller. This perspective aligns with prior work on Hybrid Zero Dynamics (HZD) [20]; however, rather than assuming stability of the zero dynamics manifold or relying on phasing variables and periodicity, we use optimal control to provably and constructively synthesize stable output-zeroing manifolds.

We propose a general framework for the control of hybrid underactuated systems and apply it to hopping, which exemplifies the challenges of such systems due to the large number of passive degrees of freedom, tight input constraints, and short ground phases. Our empirical validation of ZDPs on the ARCHER 3D hopping robot showcases an agile and stable controller as seen in Figure 1 and the supplemental video [23]. Over the course of more than 3000 hops, our method achieves state of the art disturbance rejection, hops over long distances on a treadmill, navigates an obstacle course and rough terrain without vision, and is precise enough to reliably hop across narrow bridges.

II. Preliminaries

A. Hybrid Dynamics and Lyapunov Stability

Consider an $n-$degree of freedom robotic system with coordinates $\mathbf{q} \in \mathcal{Q}$ and state $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \in \mathcal{X} \triangleq \mathsf{T}\mathcal{Q}$. Using the Euler Lagrange equations, we write the continuous-time dynamics in control-affine form as:

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \dot{\mathbf{q}} \\ -\mathbf{D}(\mathbf{q})^{-1}\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{D}(\mathbf{q})^{-1}\mathbf{B} \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} \mathbf{u} \tag{1}$$

where $\mathbf{D} : \mathcal{Q} \to \mathbb{R}^{n \times n}$ is the positive-definite mass-inertia matrix, $\mathbf{H} : \mathcal{X} \to \mathbb{R}^n$ contains the Coriolis and gravity terms, $\mathbf{B} \in \mathbb{R}^{n \times m}$ is the selection matrix, and $\mathbf{u} \in \mathbb{R}^m$ is the control input. For the following discussion we assume that $\mathbf{B}$ has (column) rank $m < n$, i.e. (1) is underactuated.

As the robot experiences impulsive effects, it is subject to the instantaneous momentum transfer equation:

$$\mathbf{x}^+ = \mathbf{\Delta}(\mathbf{x}^-), \tag{2}$$

with $\mathbf{\Delta} : \mathcal{X} \to \mathcal{X}$ representing the impact map. Combining (1) and (2), the complete hybrid dynamics can be written as:

$$\mathscr{H} = \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} & \mathbf{x} \notin \mathcal{S} \\ \mathbf{x}^+ = \mathbf{\Delta}(\mathbf{x}^-) & \mathbf{x}^- \in \mathcal{S} \end{cases}$$

where $\mathcal{S} \subset \mathcal{X}$ is an appropriately defined switching surface, for example the foot making or breaking contact with the ground [10].

Towards developing a stabilizing feedback controller for (1), define a collection of continuous time outputs $\mathbf{y} : \mathcal{X} \to \mathbb{R}^m$ that we would like to drive to zero. For outputs of relative degree two [1], consider the error coordinates $\mathbf{e} = (\mathbf{y}, \dot{\mathbf{y}}) \in \mathcal{E} \subseteq \mathbb{R}^{2m}$. These errors can be constructively stabilized via a RES-CLF, defined as:

Definition 1. [24] For the system (1), $V_\varepsilon : \mathcal{E} \to \mathbb{R}$ is said to be a *rapidly exponentially stabilizing control Lyapunov function (RES-CLF)* if there exists a $\lambda, k_1, k_2 > 0$, such that for all $\varepsilon \in (0, 1)$:

$$k_1\|\mathbf{e}\|^2 \leq V_\varepsilon(\mathbf{e}) \leq k_2\|\mathbf{e}\|^2$$
$$\inf_{\mathbf{u}} \dot{V}_\varepsilon(\mathbf{x}, \mathbf{u}) \leq -\frac{\lambda}{\varepsilon} V_\varepsilon(\mathbf{e}). \tag{3}$$

Valid relative degree ensures the existence of a nonempty set $\mathcal{K}$, defined to be the set of all controllers satisfying the inequality (3). Any controller $\mathbf{k} \in \mathcal{K}$ renders the continuous time output exponentially stable, i.e. there exists $M, \tilde{\lambda} > 0$ such that:

$$\|\mathbf{e}(t)\| \leq M e^{-\frac{\tilde{\lambda}}{\varepsilon} t} \|\mathbf{e}(0)\|,$$

whereby tuning ε down enables arbitrarily fast convergence.

B. From Hybrid Dynamics to Discrete-Time Dynamics

We will be interested in modeling $\mathscr{H}$ as a discrete-time dynamical system via its impact-to-impact dynamics. To this end, let $\mathbf{x}_k \in \mathcal{X}$ denote the robot state just before impact, P denote an admissible parameter set for $\mathbf{v}_k \in P$, a discrete parameterization of the control input over a single continuous phase, and $t_k \in \mathbb{R}_{\geq 0}$ be the duration of the continuous phase. We reformulate our hybrid control system into discrete dynamics via:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{v}_k), \tag{4}$$

where $\mathbf{F} : \mathcal{X} \times P \to \mathcal{X}$ composes the the impact map (2) with the flow of (1) under a parameterized feedback controller $\mathbf{u} = \mathbf{k}(\mathbf{x}(t), \mathbf{v}_k) \in \mathcal{K}$. In the context of hopping, we take $\mathbf{v}_k$ to be the desired impact angle. This parameterization of control input allows us to reason about the effect of impact conditions on the resulting system dynamics, which are the primary means of stabilizing legged systems. Note that here we assume the existence of a lower bound between impact times so that $\mathbf{F}$ is well defined. For a complete discussion of how to achieve this representation from the underlying hybrid dynamics, see [22]. Similar to the continuous-time case, the stability of the discrete time error dynamics can be reasoned about via Lyapunov theory:

Definition 2. For the system $\mathbf{e}_{k+1} = \mathbf{F}(\mathbf{e}_k)$, $V : \mathcal{E} \to \mathbb{R}$ is a *discrete exponential Lyapunov function* if it is positive definite and there exists an $\alpha \in (0, 1],\ k_1, k_2 > 0$ such that:

$$k_1\|\mathbf{e}_k\|^2 \leq V(\mathbf{e}_k) \leq k_2\|\mathbf{e}_k\|^2$$
$$\Delta V(\mathbf{e}) = V(\mathbf{e}_{k+1}) - V(\mathbf{e}_k) \leq -\alpha V(\mathbf{e}_k).$$

The existence of such a Lyapunov function is necessary and sufficient for exponential stability of a system, i.e. the existence of $M > 0,\ \beta \in [0, 1)$ such that:

$$\|\mathbf{e}_k\| \leq M\beta^k\|\mathbf{e}_0\|.$$

C. Discrete-Time Optimal Control

We leverage optimal control to synthesize inputs $\mathbf{v}_k$ which stabilize the discrete time system in (2) while satisfying input constraints. To this end, consider the following infinite-time optimal control problem:

$$\begin{aligned} V(\mathbf{x}_0) \triangleq \min_{\mathbf{x}_k, \mathbf{v}_k} \quad & \sum_{k=0}^{\infty} c(\mathbf{x}_k, \mathbf{v}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{v}_k) \\ & \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \leq 0 \end{aligned} \tag{5}$$

where $V : \mathcal{X} \to \mathbb{R}$ is termed the value function, $c : \mathcal{X} \times P \to \mathbb{R}$ is a positive-definite cost function and $\mathbf{h} : \mathcal{X} \times P \to \mathbb{R}^p$ contains any state-input constraints. With this, we can define the state-action value function $Q : \mathcal{X} \times P \to \mathbb{R}$ as:

$$Q(\mathbf{x}_k, \mathbf{v}_k) = c(\mathbf{x}_k, \mathbf{v}_k) + V(\mathbf{x}_{k+1}),$$

which defines the optimal control input at any state $\mathbf{x}_k$ through following optimization program:

$$\begin{aligned} \mathbf{v}_k^*(\mathbf{x}_k) = \arg\min_{\mathbf{v}_k} \ & Q(\mathbf{x}_k, \mathbf{v}_k) \\ \text{s.t.} \ & \mathbf{h}(\mathbf{v}_k, \mathbf{x}_k) \leq \mathbf{0} \end{aligned} \tag{6}$$

We rely on iteratively solving convex approximations of this nonconvex problem via iLQR. In Section III we show that tracking the output of optimal controllers in continuous time results in exponential stability of the discrete time dynamics.

D. Outputs and Zero Dynamics

Understanding the structure of underactuation provides key insight into constructing stabilizing controllers for these systems. To analyze the states that actuation directly impacts, consider the following coordinate change:

$$\boldsymbol{\eta} = \boldsymbol{\Phi}_{\boldsymbol{\eta}}(\mathbf{x}) \triangleq \begin{bmatrix} \mathbf{B}^\top \mathbf{q} \\ \mathbf{B}^\top \dot{\mathbf{q}} \end{bmatrix}, \quad \mathbf{z} = \boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{x}) \triangleq \begin{bmatrix} \mathbf{N}\mathbf{q} \\ \mathbf{N}\mathbf{D}(\mathbf{q})\dot{\mathbf{q}} \end{bmatrix} \tag{7}$$

for $\boldsymbol{\eta} \in \mathcal{N} \subset \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z} \subset \mathcal{X}$, where $\mathbf{N} \in \mathbb{R}^{(n-m)\times n}$ is chosen to be a basis for the left nullspace of $\mathbf{B}$. It is easily verified that the coordinate change $\boldsymbol{\Phi}(\mathbf{x}) \triangleq (\boldsymbol{\Phi}_{\boldsymbol{\eta}}(\mathbf{x}), \boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{x}))$ is a diffeomorphism between $\mathcal{X}$ and $\mathcal{N} \times \mathcal{Z}$; therefore, $\boldsymbol{\Phi}^{-1}$ exists and any conclusions of stability of $(\boldsymbol{\eta}, \mathbf{z})$ are directly transferable back to $\mathbf{x}$. In these coordinates, the hybrid dynamics are given by:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \hat{\mathbf{f}}(\boldsymbol{\eta}, \mathbf{z}) + \hat{\mathbf{g}}(\boldsymbol{\eta}, \mathbf{z})\mathbf{u}, \quad \dot{\mathbf{z}} = \boldsymbol{\omega}(\boldsymbol{\eta}, \mathbf{z}), \quad \boldsymbol{\Phi}^{-1}(\boldsymbol{\eta}, \mathbf{z}) \notin \mathcal{S} \\ \boldsymbol{\eta}^+ &= \boldsymbol{\Delta}_{\boldsymbol{\eta}}(\boldsymbol{\eta}^-, \mathbf{z}^-), \quad \mathbf{z}^+ = \boldsymbol{\Delta}_{\mathbf{z}}(\boldsymbol{\eta}^-, \mathbf{z}^-), \quad \boldsymbol{\Phi}^{-1}(\boldsymbol{\eta}, \mathbf{z}) \in \mathcal{S} \end{aligned}$$

termed the *actuated* dynamics and the *unactuated* dynamics, respectively. Note that these coordinates were exactly chosen such that $\hat{\mathbf{g}}(\boldsymbol{\eta}, \mathbf{z})$ is full rank and $\frac{d\mathbf{z}}{d\mathbf{x}}\mathbf{g}(\mathbf{x}) \equiv \mathbf{0}$; as such, this mapping decomposes the state space into coordinates which can directly be controlled, and those which cannot.

Assuming the continuous time input does not effect the impact map or impact time[1], applying $\boldsymbol{\Phi}$ to the discrete dynamics (4) results in:

$$\boldsymbol{\eta}_{k+1} = \hat{\mathbf{F}}(\boldsymbol{\eta}_k, \mathbf{z}_k, \mathbf{v}_k), \quad \mathbf{z}_{k+1} = \boldsymbol{\Omega}(\boldsymbol{\eta}_k, \mathbf{z}_k). \tag{8}$$

Now, consider a mapping $\boldsymbol{\psi}_{\boldsymbol{\theta}} : \mathcal{Z} \to \mathcal{N}$ and associated discrete-time error $\mathbf{e}_k = \boldsymbol{\eta}_k - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k)$. The goal will be to design $\boldsymbol{\psi}_{\boldsymbol{\theta}}$ such that driving $\mathbf{e}_k$ to zero results in stability of the overall system. This choice of error parameterization is inspired by other successful results in robotics; the Raibert Heuristic [16], reduced order models [18], and regulators for HZD gaits [21] all reason about where to place a robot's feet (the actuated state) as a function of their center of mass state (the underactuated state). We aim to generalize these methods and reason explicitly about constructive methods to generate provably stable behaviors. The construction of the mapping $\boldsymbol{\psi}_{\boldsymbol{\theta}}$ induces an associated manifold $\mathcal{M}_{\boldsymbol{\psi}} \subset \mathcal{X}$ via:

$$\mathcal{M}_{\boldsymbol{\psi}} \triangleq \{(\boldsymbol{\eta}_k, \mathbf{z}_k) \mid \boldsymbol{\eta}_k = \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k)\}. \tag{9}$$

We will be interested in enforcing conditions such that $\mathcal{M}_{\boldsymbol{\psi}}$ is controlled invariant, defined as:

Definition 3. The manifold $\mathcal{M}_{\boldsymbol{\psi}}$ is *controlled invariant* if for all $(\boldsymbol{\eta}_k, \mathbf{z}_k) \in \mathcal{M}_{\boldsymbol{\psi}}$ there exists a $\mathbf{v}_k \in P$ such that the next state remains on the manifold, i.e.:

$$\Big(\mathbf{F}(\boldsymbol{\eta}_k, \mathbf{z}_k, \mathbf{v}_k),\ \boldsymbol{\Omega}(\boldsymbol{\eta}_k, \mathbf{z}_k)\Big) \in \mathcal{M}_{\boldsymbol{\psi}}.$$

Assuming a controlled invariant manifold $\mathcal{M}_{\boldsymbol{\psi}}$, we now have the notion of discrete-time zero dynamics:

Definition 4. The *discrete-time zero dynamics* associated with a controlled invariant manifold $\mathcal{M}_{\boldsymbol{\psi}}$ are given by:

$$\mathbf{z}_{k+1} = \boldsymbol{\Omega}(\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k), \mathbf{z}_k).$$

These dynamics are autonomous but determined by choice of $\boldsymbol{\psi}_{\boldsymbol{\theta}}$; therefore, the goal of this work will be to design $\boldsymbol{\psi}_{\boldsymbol{\theta}}$ such that the zero dynamics are stable. We show that stability on $\mathcal{M}_{\boldsymbol{\psi}}$ paired with a suitably defined output controller results in stability of the overall system.

III. Discrete-Time Zero Dynamics Policies

We propose a discrete-time mapping from the underactuated state, $\mathbf{z}_k$, to a desired actuated state, $\boldsymbol{\eta}_k$. This mapping, $\boldsymbol{\psi}_{\boldsymbol{\theta}} : \mathcal{Z} \to \mathcal{N}$, will encode the desired position of the actuated coordinates given the location of the unactuated coordinates at impact. The job of the continuous time controller is to drive $\boldsymbol{\eta}(t)$ to the desired preimpact location, $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_{k+1})$.

In this section, we will first reason about the ability of continuous time controllers to render $\mathcal{M}_{\boldsymbol{\psi}}$ attractive and invariant by driving the error $\mathbf{e}$ to zero. Second, we demonstrate that if the manifold has stable zero dynamics (trajectories on the manifold converge to the origin), then stabilizing the manifold stabilizes the entire system. Finally, we propose a learning pipeline which leverages optimal control to find a manifold with the desired properties.

A. Constructive Stabilization of the Zeroing Manifold

We show that the structure of the proposed manifold allows constructive stabilization techniques:

Lemma 1. *Consider a controlled invariant manifold $\mathcal{M}_{\boldsymbol{\psi}}$. There exists a continuous-time control law $\mathbf{k} \in \mathcal{K}$ which results in exponential stabilization of $\|\boldsymbol{\eta}_k - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k)\|$.*

[1] This assumption is needed so that $\boldsymbol{\Omega}$ is not a function of $\mathbf{v}_k$ and is well justified on ARCHER as impact angle weakly effects impact time.

Proof: Consider a point $(\boldsymbol{\eta}_k, \mathbf{z}_k)$ and the evaluation of the current and next states on the manifold: $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k)$ and $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_{k+1})$, respectively. As the $\boldsymbol{\eta}(t)$ dynamics are feedback linearizable, there exists a dynamically feasible trajectory $\boldsymbol{\eta}_d(t)$ such that $\boldsymbol{\eta}_d(0) = (\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k))^+$, and $\boldsymbol{\eta}_d(t_k) = \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_{k+1})$, where t_k is the impact time and $(\cdot)^+$ denotes a postimpact state. For example, $\boldsymbol{\eta}_d(t)$ can be constructed using Bezier polynomials [25]. Using a controller $\mathbf{k} \in \mathcal{K}$, i.e. satisfying the RES-CLF condition (3), we can obtain exponential convergence to this trajectory in continuous time:

$$\|\boldsymbol{\eta}(t) - \boldsymbol{\eta}_d(t)\| \leq M e^{-\frac{\lambda}{\varepsilon} t} \|\boldsymbol{\eta}_k^+ - (\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k))^+\|,$$

for $M, \lambda > 0$. Taking $T_* > 0$ to be the lower bound between impact times, the impact states are uniformly bounded by:

$$\|\boldsymbol{\eta}_{k+1} - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_{k+1})\| \leq M e^{-\frac{\lambda}{\varepsilon} T_*} \|\boldsymbol{\eta}_k^+ - (\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k))^+\|.$$

Then, using the properties of the impact map we have:

$$\begin{aligned}\|\boldsymbol{\eta}_k^+ - (\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k))^+\| &= \|\boldsymbol{\Delta}_{\boldsymbol{\eta}}(\boldsymbol{\eta}_k, \mathbf{z}_k) - \boldsymbol{\Delta}_{\boldsymbol{\eta}}(\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k), \mathbf{z}_k)\| \\ &\leq L_\Delta \|\boldsymbol{\eta}_k - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k)\|,\end{aligned}$$

substituting into the bound above, and choosing $\varepsilon > 0$ sufficiently small that $\alpha = M L_\Delta e^{-\frac{\lambda}{\varepsilon} T_*} \in (0, 1]$, we have:

$$\|\boldsymbol{\eta}_{k+1} - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_{k+1})\| \leq \alpha \|\boldsymbol{\eta}_k - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k)\|,$$

proving exponential stability to the manifold, as desired. ■ ■

Remark 1. The desired trajectory $\boldsymbol{\eta}_d(t)$ is being implicitly replanned at impact via $\boldsymbol{\psi}_{\boldsymbol{\theta}}$ as a function of the underactuated state $\mathbf{z}_k$. Additionally, the manifold $\mathcal{M}_{\boldsymbol{\psi}}$ is invariant under the discrete dynamics $\mathbf{F}$, but is notably not hybrid invariant.

B. Composite Stability

The previous section demonstrated a method for constructing a controller to exponentially stabilize the system to a controlled invariant manifold $\mathcal{M}_{\boldsymbol{\psi}}$. We now show that exponentially stabilizing the system to a manifold with stable zero dynamics results in composite exponential stability of the entire system:

Theorem 1. *Consider a controlled invariant manifold $\mathcal{M}_{\boldsymbol{\psi}}$ whose zero dynamics are exponentially stable. Any control law exponentially stabilizing $\|\boldsymbol{\eta}_k - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k)\|$ stabilizes the discrete-time composite system $(\boldsymbol{\eta}_k, \mathbf{z}_k)$ to the origin.*

Proof: Define $\mathbf{e}_k = \boldsymbol{\eta}_k - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k)$. By Lemma 1, there exists a continuous-time controller $\mathbf{k} \in \mathcal{K}$ rendering the discrete error dynamics exponentially stable. As such, converse Lyapunov theory guarantees the existence of a Lyapunov function $V_{\mathbf{e}} : \mathcal{E} \to \mathbb{R}$ satisfying:

$$\begin{aligned}k_1\|\mathbf{e}_k\|^2 \leq V_{\mathbf{e}}(\mathbf{e}_k) &\leq k_2\|\mathbf{e}_k\|^2 \\ \Delta V_{\mathbf{e}}(\mathbf{e}_k) &\leq -k_3\|\mathbf{e}_k\|^2\end{aligned}$$

Similarly, the stability of $\mathcal{M}_{\boldsymbol{\psi}}$ implies the existence of a Lyapunov function $V_{\mathbf{z}} : \mathcal{Z} \to \mathbb{R}$ satisfying:

$$\begin{aligned}k_4\|\mathbf{z}_k\|^2 \leq V_{\mathbf{z}}(\mathbf{z}_k) &\leq k_5\|\mathbf{z}_k\|^2 \\ \Delta V_{\mathbf{z}}(\mathbf{z}_k) = V_{\mathbf{z}}(\boldsymbol{\Omega}(\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k), \mathbf{z}_k)) - V_{\mathbf{z}}(\mathbf{z}_k) &\leq -k_6\|\mathbf{z}_k\|^2\end{aligned}$$

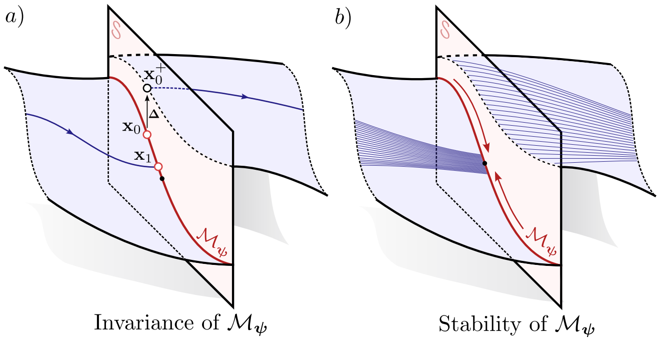
a)
b)
$\mathcal{S}$
$\mathcal{S}$
$\mathbf{x}_0^+$
Δ
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathcal{M}_{\boldsymbol{\psi}}$
$\mathcal{M}_{\boldsymbol{\psi}}$
Invariance of $\mathcal{M}_{\boldsymbol{\psi}}$
Stability of $\mathcal{M}_{\boldsymbol{\psi}}$

Fig. 2: A depiction of the two necessary properties of $\mathcal{M}_{\boldsymbol{\psi}}$: a) invariance under the discrete map $\mathbf{F}$, and b) stability.

The Lyapunov function $V_{\mathbf{z}}$ will additionally satisfy [24]:

$$|V_{\mathbf{z}}(\mathbf{z}) - V_{\mathbf{z}}(\mathbf{z}')| \leq k_7\|\mathbf{z} - \mathbf{z}'\| \left(\|\mathbf{z}\| - \|\mathbf{z}'\|\right) \triangleq \Gamma(\mathbf{z}, \mathbf{z}').$$

Consider the composite Lyapunov function candidate $V(\mathbf{e}_k, \mathbf{z}_k) \triangleq \sigma V_{\mathbf{e}}(\mathbf{e}_k) + V_{\mathbf{z}}(\mathbf{z}_k)$ with $\sigma > 0$, whereby:

$$\min\{\sigma k_1, k_4\}\|\mathbf{e}, \mathbf{z}\|^2 \leq V(\mathbf{e}, \mathbf{z}) \leq \max\{\sigma k_2, k_5\}\|\mathbf{e}, \mathbf{z}\|^2.$$

Furthermore, since $\mathbf{z}_k$ is exponentially stable on $\mathcal{M}_{\boldsymbol{\psi}}$, discrete sequences on $\mathcal{M}_{\boldsymbol{\psi}}$ will be exponentially decreasing:

$$\|\mathbf{z}_{k+1}\| = \|\boldsymbol{\Omega}(\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k), \mathbf{z}_k)\| \leq M\lambda\|\mathbf{z}_k\|,$$

for $\lambda \in [0, 1)$ and $M > 0$. Compute the difference of ΔV:

$$\begin{aligned}\Delta V &= \sigma \Delta V_{\mathbf{e}}(\mathbf{e}_k) + V_{\mathbf{z}}(\boldsymbol{\Omega}(\boldsymbol{\eta}, \mathbf{z}_k)) - V_{\mathbf{z}}(\mathbf{z}_k) \\ &= \sigma \Delta V_{\mathbf{e}}(\mathbf{e}_k) + \Delta V_{\mathbf{z}}(\mathbf{z}_k) \\ &\quad + V_{\mathbf{z}}(\boldsymbol{\Omega}(\boldsymbol{\eta}_k, \mathbf{z}_k)) - V_{\mathbf{z}}(\boldsymbol{\Omega}(\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k), \mathbf{z}_k)) \\ &\leq -\sigma k_1\|\mathbf{e}_k\|^2 - k_6\|\mathbf{z}_k\|^2 \\ &\quad + \Gamma(\boldsymbol{\Omega}(\boldsymbol{\eta}_k, \mathbf{z}_k), \boldsymbol{\Omega}(\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_k), \mathbf{z}_k)) \\ &= -\sigma k_1\|\mathbf{e}_k\|^2 - k_6\|\mathbf{z}_k\|^2 \\ &\quad + k_7 L_{\boldsymbol{\Omega}}^2\|\mathbf{e}_k\|^2 + 2M\lambda k_7 L_{\boldsymbol{\Omega}}\|\mathbf{e}_k\|\|\mathbf{z}_k\| \\ &= -\begin{bmatrix}\|\mathbf{e}_k\| \\ \|\mathbf{z}_k\|\end{bmatrix}^\top \begin{bmatrix}\frac{\sigma k_1}{2} - c(\sigma) & -M\lambda k_7 L_{\boldsymbol{\Omega}} \\ -M\lambda k_7 L_{\boldsymbol{\Omega}} & k_6\end{bmatrix} \begin{bmatrix}\|\mathbf{e}_k\| \\ \|\mathbf{z}_k\|\end{bmatrix}\end{aligned}$$

where $c(\sigma) = k_7 L_{\boldsymbol{\Omega}}^2 - \frac{\sigma}{2}k_1$, and $\Gamma(\boldsymbol{\Omega}(\boldsymbol{\eta}, \mathbf{z}), \boldsymbol{\Omega}(\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}), \mathbf{z}))$ is bounded using Lipschitz properties of the dynamics. Choosing $\sigma > \max\left\{\frac{2M^2\lambda^2 k_7^2 L_{\boldsymbol{\Omega}}^2}{k_1 k_6}, \frac{2k_7 L_{\boldsymbol{\Omega}}^2}{k_1}\right\}$ ensures the matrix is positive definite; therefore, V is a Lyapunov function certifying composite stability. ■ ■

Remark 2. Figure 2 depicts each of the assumptions used to prove stability in Theorem 1, namely discrete invariance and exponential stability of $\mathcal{M}_{\boldsymbol{\psi}}$. Subsequent sections will develop constructive techniques leveraging optimal control and learning for finding such manifolds.

C. Stability via Optimal Control

We will leverage optimality to enforce the stability on $\mathcal{M}_{\boldsymbol{\psi}}$. This choice is motivated by the fact that asymptotic stability is a necessary condition for an optimal controller to be well defined [4]. As Theorem 1 rests on assumptions of exponential stability, we define conditions under which optimality implies exponential stability:

Theorem 2. *Let $V(\mathbf{x}_k)$ be the value function for the optimal control problem defined in* (5)*, where the cost function is quadratic, $c(\mathbf{x}_k, \mathbf{v}_k) = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{v}_k^\top \mathbf{R}\mathbf{v}_k$, and the domain $\mathcal{X}$ is compact. If there exists an $\varepsilon > 0$ such that the LQR approximation of* (5) *taken by linearizing the dynamics around the equilibrium point satisfies:*

$$\mathbf{v}_{LQR}(\mathbf{x}_k) = -\mathbf{K}\mathbf{x}_k \in \mathcal{H}(\mathbf{x}_k) \quad \forall \mathbf{x}_k \in B_\varepsilon(\mathbf{0}), \tag{10}$$

with $\mathcal{H}(\mathbf{x}_k) \triangleq \{\mathbf{v}_k \in P \mid \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \leq 0\}$, then the nonlinear system is exponentially stable under the optimal controller.

Proof: We begin by showing the optimal controller (5) is exponentially stabilizing in a neighborhood of the origin. Then, we extend this claim to the entire state space. In a sufficiently small ball around the origin, LQR (10) will be exponentially stabilizing for the nonlinear system [1], as it locally satisfies input bounds. This implies constants $M_{\text{LQR}}, \delta > 0$ and $\lambda_{\text{LQR}} \in [0, 1)$ such that:

$$\|\mathbf{x}_k\| \leq M_{\text{LQR}} \lambda_{\text{LQR}}^k \|\mathbf{x}_0\| \quad \forall \mathbf{x}_0 \in B_\delta(\mathbf{0}), \;\; \forall k \in \mathbb{Z}_+.$$

We first show that the optimal trajectory emanating from an initial condition $\mathbf{x}_0 \in B_\delta(\mathbf{0})$ is similarly exponentially stable. For any $M > 0, \; \lambda \in (0, 1)$, consider two cases:

<u>*Case 1:*</u> There exists a finite index set $\{k_i\}_{i=0}^N$ satisfying:

$$\|\mathbf{x}_{k_i}\| \geq M \lambda^{k_i} \|\mathbf{x}_0\|.$$

Compute the maximum violation ratio $R \geq 1$ given by:

$$R \triangleq \max_{i \in \{0,\ldots,N\}} \frac{\|\mathbf{x}_{k_i}\|}{M \lambda^{k_i} \|\mathbf{x}_0\|}.$$

If the index set is empty, take $R = 1$. Then

$$\|\mathbf{x}_k\| \leq R M \lambda^k \|\mathbf{x}_0\| \quad \forall k \in \mathbb{Z}_+$$

And the trajectory is exponentially stable.

<u>*Case 2:*</u> There exists a infinite index set $\{k_j\}_{j=0}^\infty$ satisfying:

$$\|\mathbf{x}_{k_j}\| \geq M \lambda^{k_j} \|\mathbf{x}_0\|. \tag{11}$$

We will establish that $V(\mathbf{x}_k)$ is an exponential Lyapunov function (Definition 2) along the trajectory, and thus the trajectory is exponentially stable. First, we bound the value function difference:

$$\begin{aligned}\Delta V(\mathbf{x}_k) = V(\mathbf{x}_k) - V(\mathbf{x}_{k-1}) &= -\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k - \mathbf{v}_k^\top \mathbf{R}\mathbf{v}_k \\ &\leq -\lambda_{\min}(\mathbf{Q}) \|\mathbf{x}_k\|^2 \end{aligned} \tag{12}$$

Next, we need to show that $V(\mathbf{x}_k)$ is bounded by quadratics. Because the LQR controller is suboptimal for the nonlinear system, applying it increases the cost relative to $V(\mathbf{x}_k)$:

$$\begin{aligned} V(\mathbf{x}_0) &\leq \sum_{k=0}^{\infty} \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + (\mathbf{K}\mathbf{x}_k)^\top \mathbf{R}(\mathbf{K}\mathbf{x}_k) \\ &\leq \sum_{k=0}^{\infty} \left(\bar{\lambda}(\mathbf{Q}) + \bar{\lambda}(\mathbf{K}^\top \mathbf{R}\mathbf{K})\right) \|\mathbf{x}_k\|^2 \\ &\leq \sum_{k=0}^{\infty} \left(\bar{\lambda}(\mathbf{Q}) + \bar{\lambda}(\mathbf{K}^\top \mathbf{R}\mathbf{K})\right) M_{\text{LQR}}^2 \lambda_{\text{LQR}}^{2k} \|\mathbf{x}_0\|^2 \\ &= \frac{M_{\text{LQR}}^2}{1 - \lambda_{\text{LQR}}^2} \left(\bar{\lambda}(\mathbf{Q}) + \bar{\lambda}(\mathbf{K}^\top \mathbf{R}\mathbf{K})\right) \|\mathbf{x}_0\|^2 \end{aligned}$$

where $\underline{\lambda}$ and $\overline{\lambda}$ are the minimum and maximum eigenvalue oeprators, respectively.

Finally, using (11), we can lower bound $V(\mathbf{x}_k)$ by:

$$\begin{aligned} V(\mathbf{x}_0) &= \sum_{j=0}^{\infty} \mathbf{x}_{k_j}^\top \mathbf{Q}\mathbf{x}_{k_j} + \mathbf{v}_{k_j}^\top \mathbf{R}\mathbf{v}_{k_j} \\ &\geq \sum_{j=0}^{\infty} \underline{\lambda}(\mathbf{Q}) \|\mathbf{x}_{k_j}\|^2 \\ &\geq \sum_{j=0}^{\infty} \underline{\lambda}(\mathbf{Q}) M^2 \lambda^{2k_j} \|\mathbf{x}_0\|^2 \\ &= \left[\frac{M^2}{1 - \lambda^2} \left(\bar{\lambda}(\mathbf{Q}) + \bar{\lambda}(\mathbf{K}^\top \mathbf{R}\mathbf{K})\right) - c\right] \|\mathbf{x}_k\|^2 \end{aligned}$$

Where c is the sum of the terms removed from the geometric series. Lastly, The above bounds hold for each point on the trajectory; therefore, V is a Lyapunov function certifying exponential stability of the trajectory.

Finally, we extend the claim outside of the ball around the origin. As $V \succ 0$ and $\Delta V \prec 0$, the optimal controller is asymptotically stable [4]. By compactness of $\mathcal{X}$ and (12), the time to enter $B_\delta(\mathbf{0})$ is bounded by:

$$K \triangleq \frac{\sup_{\mathbf{x}_0 \in \mathcal{X}} V(\mathbf{x}_0)}{\inf_{\mathbf{x}_0 \in \mathcal{X} \setminus B_\delta(\mathbf{0})} \Delta V(\mathbf{x}_0)} \leq \frac{\sup_{\mathbf{x}_0 \in \mathcal{X}} V(\mathbf{x}_0)}{\underline{\lambda}(\mathbf{Q}) \delta^2}.$$

Because trajectories converge exponentially in $B_\delta(\mathbf{0})$,

$$\|\mathbf{x}_k\| \leq M \lambda^{k-K} \|\mathbf{x}_K\| \quad \forall \mathbf{x}_0 \in B_\delta(\mathbf{0}), \;\; \forall k \geq K$$

for $M > 0, \; \lambda \in [0, 1)$. By compactness of $\mathcal{X}$, trajectories are uniformly bounded $\|\mathbf{x}_k\| \leq B$; therefore:

$$\|\mathbf{x}_k\| \leq \frac{\max\{B, M\} \lambda^{-K}}{\min\{1, \delta\}} \lambda^k \|\mathbf{x}_0\| \quad \forall k \in \mathbb{Z}_+$$

is an exponential upper bound for the entire trajectory. ■■

D. Constructing the Zeroing Manifold via Learning

By Theorem 2, a manifold which is invariant under the optimal controller will be exponentially stable. Such a manifold then satisfies the assumptions of Theorem 1 and can be constructively stabilized resulting in composite stability of the entire system.

We will now present a learning method which leverages optimal control to ensure the assumptions of controlled invariance and stability of $\mathcal{M}_{\boldsymbol{\psi}}$ as depicted in Figure 2 are met. Specifically, we will search for a manifold that is invariant under the optimal action, i.e. the controller that keeps sequences of states in the manifold coincides with the optimal controller for (5).

To concisely define the loss function consider the variable

$$\boldsymbol{\zeta}_{\boldsymbol{\theta}}(\mathbf{z}) \triangleq \begin{bmatrix} \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}) \\ \mathbf{z} \end{bmatrix} \tag{13}$$

which encodes a point on the manifold. The loss function is:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\mathbf{z} \sim \text{UNIFORM}} \left\| \boldsymbol{\eta}_1^*\left(\boldsymbol{\zeta}_{\boldsymbol{\theta}}(\mathbf{z})\right) - \boldsymbol{\psi}_{\boldsymbol{\theta}}\left(\mathbf{z}_1^*\left(\boldsymbol{\zeta}_{\boldsymbol{\theta}}(\mathbf{z})\right)\right) \right\|_2^2, \tag{14}$$

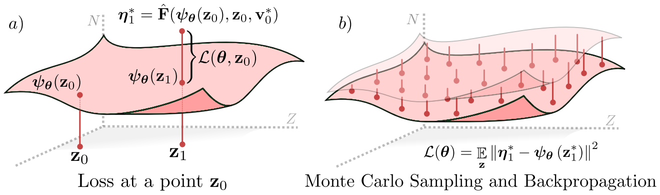
a)
$\boldsymbol{\eta}_1^* = \hat{\mathbf{F}}(\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_0), \mathbf{z}_0, \mathbf{v}_0^*)$
N
$\mathcal{L}(\boldsymbol{\theta}, \mathbf{z}_0)$
$\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_0)$
$\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_1)$
Z
$\mathbf{z}_0$
$\mathbf{z}_1$
Loss at a point $\mathbf{z}_0$
b)
N
Z
$\mathcal{L}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\mathbf{z}} \|\boldsymbol{\eta}_1^* - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_1^*)\|^2$
Monte Carlo Sampling and Backpropagation

Fig. 3: a) The loss function exactly measures the extent to which the manifold is not invariant under optimal action b) a Monte Carlo approximation of the spatial loss is used, wherein the optimal policy is backpropogated through to update the surface.

where $\mathbf{z}_1^* = \boldsymbol{\Omega}(\boldsymbol{\psi}(\mathbf{z}), \mathbf{z})$ and $\boldsymbol{\eta}_1^* = \hat{\mathbf{F}}(\boldsymbol{\psi}(\mathbf{z}), \mathbf{z}, \mathbf{v}^*)$, with $\mathbf{v}^*$ the optimal control input. The expectation is taken over a uniform distribution over $\mathcal{Z}$. The loss function directly measures how far an initial condition on the manifold deviates from the manifold under one discrete step of the optimal controller as depicted in Figure 3.

The learning pipeline outlined in Algorithm 1 starts an epoch by sampling a batch of points from $\mathcal{Z}$, therefore enabling a dimension reduction as compared to the complete state space. The network is then evaluated to produce a set of points on the current manifold, $\{\boldsymbol{\zeta}_{\boldsymbol{\theta}}(\mathbf{z}_i)\}_{i=1}^N$. We then approximately solve the optimal control problem (5). Finally, we simulate the system forwards one step to obtain $(\boldsymbol{\eta}_1^*, \mathbf{z}_1^*)$ which the loss computation in (14) requires. If $\boldsymbol{\psi}_{\boldsymbol{\theta}}$ attains zero loss, because of continuity of the network and the loss function we can conclude that the resulting manifold $\mathcal{M}_{\boldsymbol{\psi}}$ is invariant under the optimal control and can render the full order system stable by satisfaction of the preconditions for Theorem 1.

IV. Application of ZDP to ARCHER

We deployed the ZDP method on the 3D hopping robot ARCHER. To discuss the application of ZDPs to ARHCER, consider the pose of the robot $\mathbf{q} = (\mathbf{p}, q) \in \mathcal{Q}$ where $\mathbf{p} \in \mathbb{R}^3$ represents the global position in world frame and $q \in \mathbb{S}^3$ the robot's orientation quaternion. Taking the velocities to be $\mathbf{v} = (\dot{\mathbf{p}}, \boldsymbol{\omega}) \in T_{\mathbf{q}}\mathcal{Q}$ for $\dot{\mathbf{p}} \in \mathbb{R}^3$ the global linear velocity and $\boldsymbol{\omega} \in \mathfrak{s}^3$ the body frame angular rates, we can represent the full state as $\mathbf{x} = (\mathbf{q}, \mathbf{v}) \in \mathcal{X} \triangleq T\mathcal{Q}$.

ARCHER evolves under hybrid dynamics. As such, its flight and ground phase dynamics are governed by (1) and it has two impact maps of the form (2) (one for the ground to flight transition, and another for flight to ground). We treat the vertical hopping as an autonomous system, and we will focus our attention on how to stabilize the position of the robot via orientation. The flight dynamics can be decomposed into actuated states, i.e. the orientation coordinates, and unactuated states, i.e. position coordinates:

$$\boldsymbol{\eta} = \begin{bmatrix} q \\ \boldsymbol{\omega} \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{p} \\ \dot{\mathbf{p}} \end{bmatrix}.$$

Take $(\boldsymbol{\eta}_k, \mathbf{z}_k)$ to be a preimpact state. The ground phase does not depend on the control input, and the continuous-time evolution of the $\mathbf{z}$ coordinates has an extremely weak dependence on the discrete-time control input $\mathbf{v}_k$. We can assume $\boldsymbol{\Omega}$ is independent from $\mathbf{v}_k$ because the effect of different control inputs on impact time is negligible.

Algorithm 1 Monte Carlo Zero Dynamics Policy Training

```
hyperparameters: $(\Xi, \rho, \Upsilon)$
Number of MC samples, Learning Rate and Number of Steps
Initialize $\boldsymbol{\theta}$                ▷ Pretrained with reasonable policy
for $i = 1 : \Upsilon$ do
    $\mathbf{z} \sim \text{UNIFORM}(\underline{\mathbf{z}}, \overline{\mathbf{z}})$
    $\boldsymbol{\zeta}_{\boldsymbol{\theta}} \leftarrow \begin{bmatrix} \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}) \\ \mathbf{z} \end{bmatrix}$
    $\mathbf{x}_0 \leftarrow \boldsymbol{\Phi}^{-1}(\boldsymbol{\zeta}_{\boldsymbol{\theta}})$
    $\mathbf{x}_{1:T}^*, \mathbf{v}_{1:T}^* \leftarrow \text{iLQR}(\mathbf{x}_0)$
    $\begin{bmatrix} \boldsymbol{\eta}_1^*(\boldsymbol{\zeta}_{\boldsymbol{\theta}}(\mathbf{z})) \\ \mathbf{z}_1^*(\boldsymbol{\zeta}_{\boldsymbol{\theta}}(\mathbf{z})) \end{bmatrix} \leftarrow \boldsymbol{\Phi}(\mathbf{x}_1)$
    $\boldsymbol{\theta}_{i+1} \leftarrow \boldsymbol{\theta}_i - \rho \nabla_{\boldsymbol{\theta}} \sum_{\mathbf{z}} \|\boldsymbol{\eta}_1^*(\boldsymbol{\zeta}_{\boldsymbol{\theta}}(\mathbf{z})) - \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}_1^*(\boldsymbol{\zeta}_{\boldsymbol{\theta}}(\mathbf{z})))\|_2^2$
end for
return $\boldsymbol{\theta}$
```

A. *Online Control Implementation*

Given a function $\boldsymbol{\psi}_{\boldsymbol{\theta}}$, the controller aims to stabilize its associated zeroing manifold $\mathcal{M}_{\boldsymbol{\psi}}$. Consider a state $(\boldsymbol{\eta}(t), \mathbf{z}(t))$ during the flight phase. We set the desired orientation to $\boldsymbol{\eta}_d(t) = \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{z}(t))$, and update this continuously throughout the flight phase. The desired set point is converted to a quaternion, q_d, which we stabilize using the following quaternion PD controller in the flight phase:

$$\mathbf{u} = -\mathbf{K}_p \log(q_d^{-1} q) - \mathbf{K}_d \omega,$$

for suitable gains $\mathbf{K}_p, \mathbf{K}_d$. This controller is applied at 1kHz.

One key addition to the controller as compared to previous work [26] is the application of flywheel spindown in the ground phase. When the robot is in contact with the floor, the following control action is applied:

$$\mathbf{u} = -\gamma \dot{\boldsymbol{\vartheta}},$$

where $\dot{\boldsymbol{\vartheta}} \in \mathbb{R}^3$ represents the flywheel speed. This allows the system to maintain lower flywheel speeds and mitigates the problem of speed-torque constraints. This ground phase controller preserves the theoretical assumptions since the ground phase control is independent of output of the policy.

There are a few implementation differences from our theoretical implementation. The controller used in the proof of Lemma 1 differs from ours by (1) predicting the preimpact state $\mathbf{z}_{k+1}$, (2) tracking a trajectory $\boldsymbol{\eta}_d(t)$ defined by a bezier polynomial, and (3), using a RES-CLF. Empirically, a well tuned PD controller was sufficient to stabilize the continuous time system, and the feedforward input tracking that a trajectory would provide was not necessary.

B. *ZDP Optimization and Learning Details*

Notice that for discrete-time systems, (5) is a nonlinear program even if the value function is available. To solve this optimal control problem, we employ Iterative LQR (iLQR), subject to box input constraints [27]. The iLQR problem is solved in the $\mathbf{x}$ variable, so the initial condition is obtain via $\mathbf{x} = \boldsymbol{\Phi}^{-1}(\boldsymbol{\eta}, \mathbf{z})$. We implemented Algorithm 1 in the JAX [28] and used a Network of 2 Layers with 256 hidden units each using ReLu activations. In our implementation of iLQR, we assume that the low-level controller has perfect tracking and exactly achieves the desired angle with zero angular

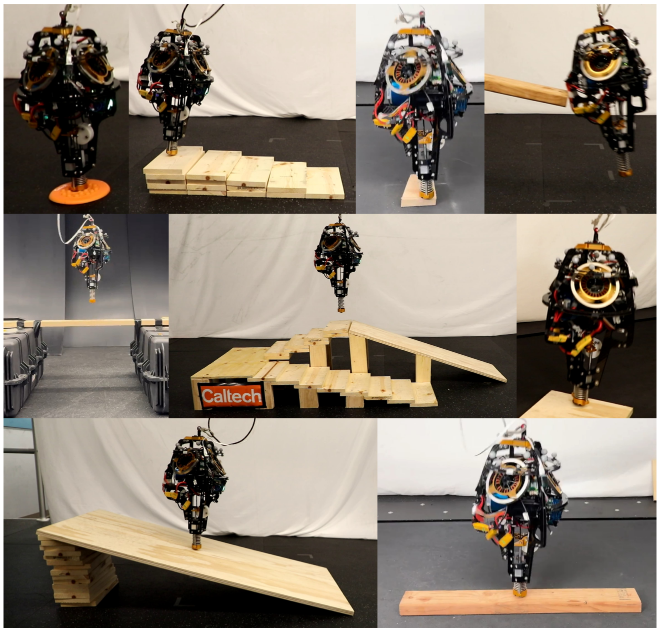

Fig. 4: A snapshot of the experiments conducted with ARCHER, including set point tracking, disturbance rejection, and hopping over rough terrain.

velocity. This considerably simplifies the flight dynamics and therefore the trajectory optimization, allowing them to be solved for in closed form. The input bounds $\mathcal{H}(\mathbf{x}_k)$ were chosen such that the torque applied during flight is bounded by the difference between the post-impact state and the desired preimpact state. We require gradients of the optimal control, $\frac{d\mathbf{v}}{d\mathbf{x}}$, as presented in [29] – note that if no constraints are active, then this gradient is exactly the feedback matrix $\mathbf{K} = \mathbf{Q}_{\mathbf{vv}}^{-1}\mathbf{Q}_{\mathbf{vx}}$ from the iLQR algorithm.

iLQR requires a stabilizing initial guess in order to converge; therefore, we use a Raibert heuristic for the first rollout. To eliminate this dependence, other optimal control methods could be used, for instance SQP. The authors experienced difficulty with the speed and accuracy of large-scale QP solvers in JAX and leveraged the fact that iLQR solves many small QPs for speed and stability. Additionally, for computational efficiency, we limit the number of iLQR iterations to five (empirically enough to obtain convergence for this system). The full code base for this project can be found at [30].

V. Results and Limitations

A. Hardware Results

A collection of the experiments conducted on ARCHER can be seen in Figure 4. The ARCHER hardware platform [31] consists of three KV115 T-Motors with 250 g flywheel masses attached for orientation control, and one U10-plus T-Motor attached to a 3-1 gear reduction to the foot via a cable and pulley system. The robot is powered by two 6 cell LiPo betteries connected in series, which can supply up to 50.8 V at over 100 A of current to the four ELMO Gold Solo Twitter motor controllers. The policy $\psi_{\boldsymbol{\theta}}$ was exported from JAX to an ONNX file, which is evaluated at 1kHz on an Ubuntu 20.04 machine with AMD Ryzen 5950x @ 3.4 GHz and 64 Gb RAM and torques are passed directly to the robot over ethernet. This controller does not require this amount of compute to run, and could be feasibly implemented on an NVIDIA Jetson or comparable board. A Kalman filter with projectile dynamics is used to filter the position estimates from optritrack in the flight phase. The manif library [32] is used to compute the $\log$ map for the quaternion PD controller.

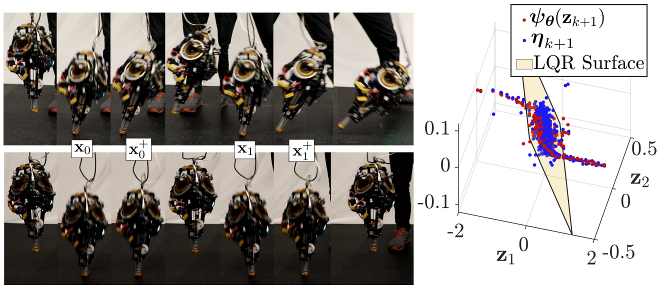

Fig. 5: Left: A comparison between LQR (top) and ZDPs (bottom) while tracking a 2 m setpoint. Right: The output of the trained policy and the actual state at impact over 3000 hops, as compared to an LQR controller.

We logged over 3,000 stable hops when deploying the ZDP method on the ARCHER hardware platform, a selection of which can be seen in Figure 4 and in the supplemental video [23]. Figure 5 depicts the desired impact angle, i.e. the learned policy evaluation, and the actual impact angle over the complete collection of all hardware tests. In general, as predicted by the theory, this manifold is both invariant under the feedback controller, and stable. Also interesting to note is that around the origin, the learned policy alignes with LQR, as presented in Theorem 2. Notably, away from the origin, the learned policy diverges from LQR in order to maintain stability under the enforced input contstraints. A comparison between the trained policy and the application of a naive LQR controller when trying to track a setpoint 2 m away is seen in the left part of Figure 5, wherein ZDPs maintain stability by implicitly enforcing discrete invariance and optimality over a horizon.

The tight trajectory tracking and system behavior is seen in Figure 6, where ARCHER was asked to follow two laps of a 1 m square trajectory. As seen on the right of Figure 6, using a PD controller at the feedback level empirically resulted in the error (and therefore the torques) converging exponentially fast to a small neighborhood of zero during the flight phase. During this torque application, the flywheel speed can be seen to grow, while the ground phase controller is able to successfully regulate them close to zero.

B. Limitations

As training this policy involves querying the optimal control input and its gradients, each iteration of the training process is computationally expensive (2 seconds per iteration for a batch size of 30). The use of iLQR required a stabilizing controller to initialize the rollout and therefore can only do local improvements on a stabilizing policy. Furthermore, to avoid sampling initial conditions in the training pipeline which the hopper cannot stabilize, the policy $\psi_{\boldsymbol{\theta}}$ was pre-trained with a conservative Raibert heuristic.

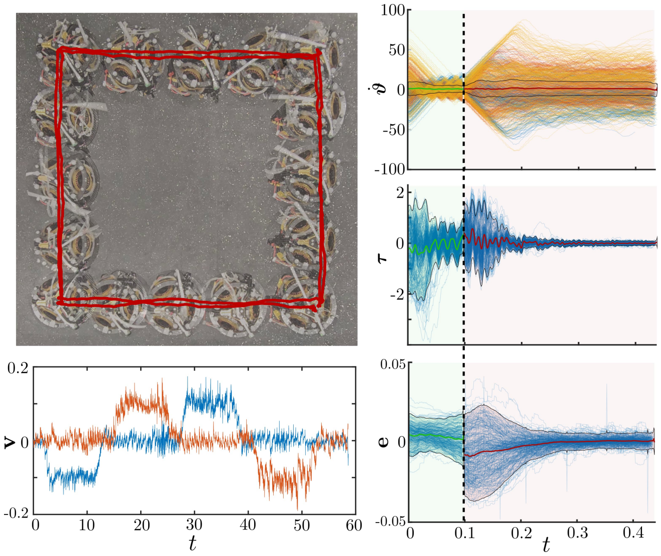

Fig. 6: Square trajectory tracking. Left pane: overhead view with positional hardware data overlayed (top) and velocity tracking (bottom). Right pane: wheel velocities (top), torque (mid), and error (bottom) in the ground (green) and flight (red) phase with mean and 2σ deviation.

VI. Conclusion and Future Work

We have proposed a method of synthesizing stabilizing feedback controllers for hybrid underactuated systems. By exploiting the zero dynamics decomposition, we demonstrated both theoretically and experimentally that stabilizing such systems can effectively be decomposed into designing a mapping which renders the discrete zeroing manifold invariant under optimal controllers and pairing it with a suitable tracking controller. Future work includes merging the proposed methods with RL controllers, applying to other legged systems, and developing a parallel theory for continuous time systems.

VII. Acknowledgements

We would like to thank Murtaza Hathiyari for aiding with C++ code development and hardware experiment testing.

References

[1] S. Sastry, "Linearization by State Feedback," in *Nonlinear Systems: Analysis, Stability, and Control*, ser. Interdisciplinary Applied Mathematics, S. Sastry, Ed. Springer, 1999, pp. 384–448.

[2] I. D. J. Rodriguez, N. Csomay-Shanklin, Y. Yue, and A. D. Ames, "Neural gaits: Learning bipedal locomotion via control barrier functions and zero dynamics policies," in *Proceedings of The 4th Annual L4DC*, vol. 168. PMLR, Jun 2022, pp. 1060–1072.

[3] W. Compton, I. D. J. Rodriguez, N. Csomay-Shanklin, Y. Yue, and A. D. Ames, "Constructive nonlinear control of underactuated systems via zero dynamics policies," *preprint arXiv:2408.14749*, 2024.

[4] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2012.

[5] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[6] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[7] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, "Optimization-based control for dynamic legged robots," *Trans. Rob.*, vol. 40, p. 43–63, oct 2023.

[8] C. Khazoom, S. Hong, M. Chignoli, E. Stanger-Jones, and S. Kim, "Tailoring solution accuracy for fast whole-body model predictive control of legged robots," *preprint arXiv:2407.10789*, 2024.

[9] H. Li and P. M. Wensing, "Cafe-mpc: A cascaded-fidelity model predictive control framework with tuning-free whole-body control," *preprint arXiv:2403.03995*, 2024.

[10] E. Westervelt, J. Grizzle, and D. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, Jan. 2003.

[11] J. Reher, "Dynamic bipedal locomotion: From hybrid zero dynamics to control lyapunov functions via experimentally realizable methods," Ph.D. dissertation, California Institute of Technology, 2021.

[12] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proceedings of ICLR*, 2016.

[13] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[14] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *preprint arXiv:2401.16889*, 2024.

[15] H. J. Suh, M. Simchowitz, K. Zhang, and R. Tedrake, "Do differentiable simulators give better policy gradients?" in *ICML*. PMLR, 2022, pp. 20 668–20 696.

[16] M. H. Raibert, H. B. Brown, and M. Chepponis, "Experiments in Balance with a 3D One-Legged Hopping Machine," *IJRR*, vol. 3, no. 2, pp. 75–92, Jun. 1984, publisher: SAGE Publications Ltd STM.

[17] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proceedings 2001 IEEE/RSJ ICIRS (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 239–246.

[18] B. Han, H. Yi, Z. Xu, X. Yang, and X. Luo, "3d-slip model based dynamic stability strategy for legged robots with impact disturbance rejection," *Scientific Reports*, vol. 12, no. 1, p. 5892, 2022.

[19] A. Isidori, "Elementary Theory of Nonlinear Feedback for Single-Input Single-Output Systems," in *Nonlinear Control Systems*, ser. Communications and Control Engineering. London: Springer, 1995, pp. 137–217.

[20] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[21] J. Reher and A. D. Ames, "Control lyapunov functions for compliant hybrid zero dynamic walking," *preprint arXiv:2107.04241*, 2021.

[22] X. Da and J. Grizzle, "Combining trajectory optimization, supervised machine learning, and model structure for mitigating the curse of dimensionality in the control of bipedal robots," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1063–1097, 2019.

[23] "Supplemental video." [Online]. Available: https://vimeo.com/923800815

[24] A. D. Ames and I. Poulakakis, "Hybrid zero dynamics control of legged robots," *Bioinspired Legged Locomotion: Models, Concepts, Control and Applications*, pp. 292–331, 2017.

[25] N. Csomay-Shanklin, A. J. Taylor, U. Rosolia, and A. D. Ames, "Multi-rate planning and control of uncertain nonlinear systems: Model predictive control and control lyapunov functions," in *2022 IEEE 61st CDC*. IEEE, 2022, pp. 3732–3739.

[26] N. Csomay-Shanklin, V. D. Dorobantu, and A. D. Ames, "Nonlinear Model Predictive Control of a 3D Hopping Robot: Leveraging Lie Group Integrators for Dynamically Stable Behaviors," in *2023 ICRA*. London, United Kingdom: IEEE, May 2023, pp. 12 106–12 112.

[27] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 ICRA*. IEEE, 2014, pp. 1168–1175.

[28] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.

[29] B. Amos, I. Jimenez, J. Sacks, B. Boots, and J. Z. Kolter, "Differentiable mpc for end-to-end planning and control," *Advances in neural information processing systems*, vol. 31, 2018.

[30] "Code," 2024. [Online]. Available: https://github.com/ivandariojr/LearnedZeroDynamicsPolicies

[31] E. R. Ambrose, "Creating ARCHER: A 3D Hopping Robot with Flywheels for Attitude Control," Ph.D. dissertation, California Institute of Technology, 2022.

[32] J. Deray and J. Solà, "Manif: A micro Lie theory library for state estimation in robotics applications," *Journal of Open Source Software*, vol. 5, no. 46, p. 1371, 2020.