

HALO: Hybrid Auto-encoded Locomotion with Learned Latent Dynamics, Poincaré Maps, and Regions of Attraction

Blake Werner*

Sergio Esteban*

Massimiliano de Sa

Caltech, Pasadena, CA, USA

Max H. Cohen

NC State University, Raleigh, NC, USA

Aaron D. Ames

Caltech, Pasadena, CA, USA

BWERNER@CALTECH.EDU

SESTEBAN@CALTECH.EDU

MDESA@CALTECH.EDU

MHCOHEN2@NCSTU.EDU

AMES@CALTECH.EDU

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

Reduced-order models are powerful for analyzing and controlling high-dimensional dynamical systems. Yet constructing these models for complex hybrid systems such as legged robots remains challenging. Classical approaches rely on hand-designed template models (e.g., LIP, SLIP), which, though insightful, only approximate the underlying dynamics. In contrast, data-driven methods can extract more accurate low-dimensional representations, but it remains unclear when stability and safety properties observed in the latent space meaningfully transfer back to the full-order system. To bridge this gap, we introduce HALO (Hybrid Auto-encoded Locomotion), a framework for learning latent reduced-order models of periodic hybrid dynamics directly from trajectory data. HALO employs an autoencoder to identify a low-dimensional latent state together with a learned latent Poincaré map that captures step-to-step locomotion dynamics. This enables Lyapunov analysis and the construction of an associated region of attraction in the latent space, both of which can be lifted back to the full-order state space through the decoder. Experiments on a simulated hopping robot and full-body humanoid locomotion demonstrate that HALO yields low-dimensional models that retain meaningful stability structure and predict full-order region-of-attraction boundaries.

Keywords: Autoencoders, Hybrid Systems, Reduced-Order Models.

1. Introduction

Reduced-order model (ROM) techniques aim to approximate high-dimensional dynamical systems with lower-dimensional representations. Such reductions enable more efficient verification, control synthesis, and planning for systems whose full-order model (FOM) dynamics may be too complex to analyze directly. Classical approaches to model reduction in control include balanced truncation and related projection-based methods [Antoulas \(2005\)](#); [Benner et al. \(2015\)](#), moment-matching for linear and nonlinear systems [Astolfi \(2010\)](#), and methods based on invariant manifolds [Haller and Ponsioen \(2016\)](#). These approaches share conceptual ties with notions of abstraction in hybrid and symbolic control [Tabuada \(2009\)](#); [Girard and Pappas \(2009\)](#), where simpler surrogate systems are constructed to preserve relevant dynamical properties of the original one.

While these methods provide rigorous guarantees for structured or weakly nonlinear systems, extending them to high-dimensional robotic systems, particularly those with contact-rich dynamics, remains challenging. Motivated by these limitations, recent works learn ROMs from data.

Approaches based on dynamic mode decomposition, PCA, and proper orthogonal decomposition [Brunton and Kutz \(2022\)](#) identify dominant low-dimensional structure in trajectories, while autoencoders learn nonlinear embeddings that map between full-order states and latent coordinates. A central challenge, however, is determining when properties such as stability, safety, or invariance in the latent space faithfully transfer back to the original full-order dynamics. Recent work has begun to establish sufficient conditions and approximation guarantees in this direction [Lutkus et al. \(2025\)](#); [Nakamura et al. \(2025\)](#), though practical instantiations for hybrid locomotion remain limited.

ROMs play a particularly influential role in legged robotics, where simple models such as the linear inverted pendulum (LIP) [Kajita et al. \(2001\)](#) and the spring-loaded inverted pendulum (SLIP) [Blickhan \(1989\)](#) capture the salient dynamics of center-of-mass motion [Raibert \(1986\)](#); [Wensing and Revzen \(2017\)](#). These templates are effective for control design and gait stabilization, but are inherently approximate: they do not directly arise from a systematic reduction of the FOM, and their accuracy can degrade outside of nominal operating regimes. Thus, recent work has learned reduced-order locomotion models from data [Castillo et al. \(2024\)](#); [Chen et al. \(2024a,b\)](#); [Li et al. \(2024\)](#); [Starke et al. \(2022\)](#), though questions remain regarding how to analyze their stability properties.

Overview and Contributions: In this paper, we propose HALO (Hybrid Autoencoded Locomotion), a framework for constructing and analyzing ROMs of hybrid robotic locomotion using autoencoder-based latent representations. Our approach represents the high-dimensional hybrid dynamics as a discrete-time system via a Poincaré return map [Westervelt et al. \(2018\)](#). We then learn an autoencoder that embeds this return map into a low-dimensional latent space, yielding a latent Poincaré map that captures the step-to-step evolution of the locomotion gait cycle.

We first motivate this reduction in the linear setting, then extend to the nonlinear case by leveraging the existence of low-dimensional invariant surfaces associated with periodic orbits. This perspective provides practical guidance for selecting the latent dimension and clarifies when the autoencoder can represent the orbit geometry with negligible reconstruction error. Once the latent dynamics are identified, we perform Lyapunov stability analysis directly in the latent space and subsequently lift the corresponding region of attraction to the full-order state space via the decoder.

We evaluate HALO across a suite of hybrid systems. Across all systems, we show that stability properties inferred from the latent Poincaré dynamics predict stability in the full-order system, even when it is controlled via reinforcement learning policies.

2. Preliminaries

Notation: The Euclidean and Frobenius norms are denoted by $\|\cdot\|$ and $\|\cdot\|_F$, respectively. The restriction of a function $f : A \rightarrow B$ to a subset $U \subseteq A$ is denoted $f|_U : U \rightarrow B$. The inclusion map of a subset $U \subseteq A$ into A , which takes each $u \in U$ to $u \in A$, is denoted $\iota_U : U \hookrightarrow A$.

Model Reduction & Autoencoders: High-dimensional dynamical systems often have an underlying low-dimensional structure. We formalize extracting low-dimensional structure from dynamical systems as follows. Consider a discrete-time dynamical system:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k), \quad \mathbf{x}_k \in \mathbb{R}^{n_x}, \quad \mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}. \quad (1)$$

We refer to $\mathbf{x}_k \in \mathbb{R}^{n_x}$ as the *full-order state*, and \mathbf{f} as the FOM. To obtain a ROM, we must *compress* \mathbf{x} to a lower-dimensional space, and \mathbf{f} to a lower-dimensional model. Additionally, we must have a way to *decompress* from this reduced state back to \mathbf{x} , motivating the use of *autoencoders*.

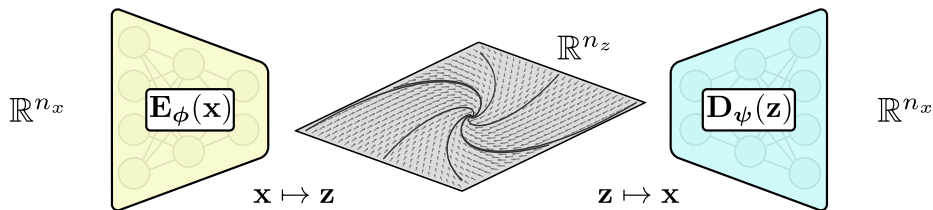


Figure 1: Autoencoders enable learning of a reduced-order dynamics model in a latent space.

The process of compression and decompression is performed by *encoder* and *decoder* maps. First, we define $n_z < n_x$ to be the dimension of our ROM. An encoder $\mathbf{E} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$ performs compression, and a decoder $\mathbf{D} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_x}$ performs decompression. Since \mathbb{R}^{n_z} represents the hidden low-dimensional structure, we refer to \mathbb{R}^{n_z} as the *latent space*, and n_z as the *latent dimension*. We say an encoder-decoder pair (\mathbf{E}, \mathbf{D}) is *perfect* if $\mathbf{D} \circ \mathbf{E}(\mathbf{x}) = \mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^{n_x}$, i.e., no information is lost from compression to decompression. In this spirit, to find a low-dimensional representation of \mathbf{f} , we seek a map $\mathbf{g} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_z}$ that captures the important features of \mathbf{f} . We refer to the map $\mathbf{g} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_z}$ as the *reduced-order model*. This model determines a recurrence:

$$\mathbf{z}_{k+1} = \mathbf{g}(\mathbf{z}_k), \quad \mathbf{z}_k \in \mathbb{R}^{n_z}, \quad (2)$$

in the latent space. In a perfect world, one would seek a perfect encoder-decoder pair (\mathbf{E}, \mathbf{D}) for which $\mathbf{f}(\mathbf{x}) = \mathbf{D} \circ \mathbf{g} \circ \mathbf{E}(\mathbf{x})$, for all $\mathbf{x} \in \mathbb{R}^{n_x}$. Such an encoder-dynamics pair would perfectly compress the original high-order system down to a low-dimensional latent space.

For complex, real-world systems, perfect compression in this sense is unrealistic. Likewise, one cannot analytically determine optimal encoders, decoders, and ROMs. To find a “good” encoder-decoder and ROM, one practically represents each by a neural network $\mathbf{E}_\phi, \mathbf{D}_\psi$, with parameters ϕ and ψ , respectively, and encodes the goals above in appropriately designed *loss functions*.

From Discrete to Continuous with Hybrid Systems: Above, we discussed the construction of autoencoder models for discrete-time dynamical systems. In robotics, however, there are a wide class of systems that do not fall naturally into this category. Robotic systems, such as legged robots, are *hybrid systems* Westervelt et al. (2018) that exhibit both discrete *and* continuous-time behavior:

$$\mathcal{H} = \begin{cases} \dot{\mathbf{x}} = \mathbf{f}_c(\mathbf{x}), & \mathbf{x} \in \mathbb{R}^{n_x} \setminus \mathcal{S} \\ \mathbf{x}^+ = \mathbf{\Delta}(\mathbf{x}^-), & \mathbf{x}^- \in \mathcal{S}, \end{cases} \quad (3)$$

where $\mathcal{S} \subseteq \mathbb{R}^{n_x}$ is a set that triggers a discrete change in behavior. For legged robots, this change is triggered by a foot strike with the ground. Under certain conditions, one may construct a map $\mathbf{f} : \mathcal{S} \rightarrow \mathcal{S}$, termed the *Poincaré map* that maps a state $\mathbf{x} \in \mathcal{S}$ to the state $\mathbf{f}(\mathbf{x}) \in \mathcal{S}$ at which the system next intersects \mathcal{S} ; for a legged robot, \mathbf{f} maps from the state at one foot strike to the state at the next foot strike. Thus, the Poincaré map induces a discrete-time system $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$ capturing the step-to-step dynamics. Studying the stability of the Poincaré map affords insights into the behavior of the hybrid system. Details on hybrid systems and Poincaré maps are provided in Appendix A.

3. Insights from Linear and Nonlinear Systems

In the previous section, we discussed autoencoders, ROMs, and Poincaré maps. Autoencoders provide a means to compute ROMs from complex, high-dimensional data; ROMs enable analyzing

complex systems using simple models; Poincaré maps yield insight into the stability of hybrid systems from simpler discrete-time models. We now bring these threads together to study the behavior of hybrid systems. We use autoencoders to learn ROMs of Poincaré maps and analyze their stability in the latent space to infer stability properties of the full hybrid system. We begin by studying simple yet insightful motivation from linear and nonlinear dynamics, which provide templates for our learning-based approach. Then, we outline an autoencoder method for learning ROMs of Poincaré maps. We defer all proofs in this section to Appendix B.

Motivation from Linear Systems: To motivate the nonlinear problem, we first perform a simple study for linear systems. Consider a full-order model governed by discrete-time LTI dynamics $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k$, where $\mathbf{x}_k \in \mathbb{R}^{n_x}$. Our goal is to reduce the system to a lower-dimensional linear space. To this end, we fix a *reduced dimension* n_z , smaller than $\text{rank}(\mathbf{A})$. To perform dimensionality reduction, we define three matrices: the *encoder matrix* $\mathbf{E} \in \mathbb{R}^{n_z \times n_x}$, the *decoder matrix* $\mathbf{D} \in \mathbb{R}^{n_x \times n_z}$, and the *latent dynamics matrix* $\mathbf{Q} \in \mathbb{R}^{n_z \times n_z}$. How do we pick $\mathbf{D}, \mathbf{Q}, \mathbf{E}$ such that trajectories of the autoencoder model, $\hat{\mathbf{x}}_k = \mathbf{D}\mathbf{Q}^k\mathbf{E}\mathbf{x}_0$, resemble those of the full-order model? First, we consider the problem of matching *one step* of the dynamics. Using the Eckart-Young theorem [Calafiore and El Ghaoui \(2014\)](#), we solve for the $\mathbf{E}, \mathbf{D}, \mathbf{Q}$ minimizing the worst-case one-step error.

Lemma 1 (One-Step Optimal Matching) *Let $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ and $n_z \in (0, \text{rank}(\mathbf{A}))_{\mathbb{Z}}$. Consider:*

$$\inf_{\mathbf{E} \in \mathbb{R}^{n_z \times n_x}, \mathbf{D} \in \mathbb{R}^{n_x \times n_z}, \mathbf{Q} \in \mathbb{R}^{n_z \times n_z}} \sup_{\mathbf{x}_0 \in \mathbb{R}^{n_x}, \|\mathbf{x}_0\|=1} \|\mathbf{A}\mathbf{x}_0 - \mathbf{D}\mathbf{Q}\mathbf{E}\mathbf{x}_0\|^2. \quad (4)$$

If $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is an SVD of \mathbf{A} , then $\mathbf{D} = \mathbf{U}_{n_z}$, $\mathbf{Q} = \mathbf{\Sigma}_{n_z}$, $\mathbf{E} = \mathbf{V}_{n_z}^\top$, where $\mathbf{U}_{n_z}, \mathbf{V}_{n_z}$ are the first n_z columns of \mathbf{U}, \mathbf{V} and $\mathbf{\Sigma}_{n_z} = \text{diag}(\sigma_1, \dots, \sigma_{n_z})$, are optimal solutions to (4).

Now, we consider the multi-step version of Lemma 1. We seek a problem with a simple, closed-form solution that provides insight into the more general autoencoder problem. Inspired by the \mathcal{H}_2 model reduction problem [Dullerud and Paganini \(2013\)](#), we consider the optimization problem:

$$\inf_{\mathbf{E} \in \mathbb{R}^{n_z \times n_x}, \mathbf{D} \in \mathbb{R}^{n_x \times n_z}, \mathbf{Q} \in \mathbb{R}^{n_z \times n_z}} \sum_{k=0}^{\infty} \left\| \mathbf{A}^k - \mathbf{D}\mathbf{Q}^k\mathbf{E} \right\|_F^2. \quad (5)$$

Though this problem is not generally solvable in closed-form, there are certain special cases for which closed-form solutions exist and are informative.

Lemma 2 (Diagonal Model Reduction) *Suppose $\mathbf{A} = \text{diag}(\lambda_1, \dots, \lambda_{n_x})$, with $|\lambda_1| \geq \dots \geq |\lambda_{n_x}|$. Let $n_z \in (0, \text{rank}(\mathbf{A}))_{\mathbb{Z}}$. If the unstable and center subspaces of \mathbf{A} together have dimension $\leq n_z$, then (5) is feasible and an optimal solution is given by the n_z -truncation:*

$$\hat{\mathbf{D}} = \begin{bmatrix} I_{n_z} \\ 0_{(n_x-n_z) \times n_z} \end{bmatrix}, \quad \hat{\mathbf{Q}} = \text{diag}(\lambda_1, \dots, \lambda_{n_z}), \quad \hat{\mathbf{E}} = [I_{n_z} \quad 0_{n_z \times (n_x-n_z)}]. \quad (6)$$

Lemma 2 suggests a heuristic for viewing the nonlinear dimensionality reduction problem: if a latent dimension is capable of capturing complex, unstable system behavior, and all remaining modes are stable, then the latent dimension is capable of producing a “good” reduced-order model. In what follows, we further develop this perspective in the nonlinear setting.



Figure 2: An *attractive invariant manifold* for a discrete-time system (in red) can have a lower dimensional embedding (in blue). This embedding informs a choice of latent dimension and reduced-order model.

Motivation from Nonlinear Systems: To begin our nonlinear analysis, recall the *manifold hypothesis*, which posits that data points in real-world, high-dimensional data sets are often clustered around low-dimensional *submanifolds* Fefferman et al. (2016). Now, we apply this hypothesis, which underlies various dimensionality reduction methods, to our model reduction problem.

Suppose we collect trajectory data $\mathcal{D} = \{\{\mathbf{x}_k\}_{k=0}^K : \mathbf{x}_0 \in \mathcal{X} \subseteq \mathbb{R}^n, \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)\}$ from a variety of initial conditions $\mathbf{x}_0 \in \mathcal{X} \subseteq \mathbb{R}^n$. By the manifold hypothesis, suppose the trajectory data concentrates around a submanifold of \mathbb{R}^n . Since we sample \mathbf{x}_0 from across the state space, trajectories clustering around a submanifold suggest the existence of an *attractive invariant manifold* to which nearby trajectories converge. To gain insight into the nonlinear autoencoder problem, we thus present a simple study relating invariant manifolds and autoencoder reduced-order models.

First, employing the classic *Whitney embedding theorem* (Lee, 2013, Thm. 6.19), we establish a basic relationship between the dimensionality of an invariant manifold \mathcal{M} (for which $\mathbf{f}(\mathcal{M}) = \mathcal{M}$) and the dimension of latent space required to fully represent \mathcal{M} and the dynamics on it.

Lemma 3 (Model Reduction & Invariant Manifolds) *Let $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ be a map defining a discrete-time system with smooth invariant manifold $\mathcal{M} \subseteq \mathbb{R}^{n_x}$. For $\dim(\mathcal{M}) = k$:*

1. Perfect encoder-decoder pair on \mathcal{M} : *there exists a smooth encoder $\mathbf{E} : U \subseteq \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$ and a smooth decoder $\mathbf{D} : W \subseteq \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_x}$ satisfying $\mathbf{D}(\mathbf{E}(\mathbf{x})) = \mathbf{x}$ for all $\mathbf{x} \in \mathcal{M}$, where $n_z \leq 2k$ and U, W are open neighborhoods of \mathcal{M} and $\mathbf{E}(\mathcal{M})$ respectively,*
2. Latent system: *there exists a map $\mathbf{g} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_z}$ satisfying $\mathbf{f}(\mathbf{x}) = \mathbf{D} \circ \mathbf{g} \circ \mathbf{E}(\mathbf{x}) \forall \mathbf{x} \in \mathcal{M}$.*

Remark 4 *Further reductions in n_z are possible assuming compactness of \mathcal{M} and analogues of (1) and (2) on open subsets of \mathcal{M} instead of on its entirety Kvalheim and Sontag (2025).*

Using \mathbf{E} and \mathbf{D} constructed in the proof of Lemma 3, we make the simple observation that stability of the latent dynamics on $\mathbf{E}(\mathcal{M})$ implies stability of the original system on \mathcal{M} .

Theorem 5 (Transfer of Stability on \mathcal{M}) *Consider the setting of Lemma 3 and its proof. Let $\tilde{\mathbf{g}} : \mathbf{E}(\mathcal{M}) \rightarrow \mathbf{E}(\mathcal{M})$ be the restriction of \mathbf{g} to $\mathbf{E}(\mathcal{M})$ and $\tilde{\mathbf{f}} : \mathcal{M} \rightarrow \mathcal{M}$ the restriction of \mathbf{f} to \mathcal{M} . Suppose $\mathbf{z}^* \in \mathbf{E}(\mathcal{M})$ is a locally asymptotically stable fixed point of $\tilde{\mathbf{g}}$. Then, $\mathbf{x}^* = \mathbf{D}(\mathbf{z}^*)$ is a locally asymptotically stable fixed point of $\tilde{\mathbf{f}}$. Further, if \mathbf{z} belongs to the region of attraction of \mathbf{z}^* for $\tilde{\mathbf{g}}$, then $\mathbf{D}(\mathbf{z})$ belongs to the region of attraction of \mathbf{x}^* for $\tilde{\mathbf{f}}$.*

Remark 6 *Here, stability is with respect to open subsets of $\mathbf{E}(\mathcal{M})$ and \mathcal{M} , not \mathbb{R}^{n_z} and \mathbb{R}^{n_x} .*

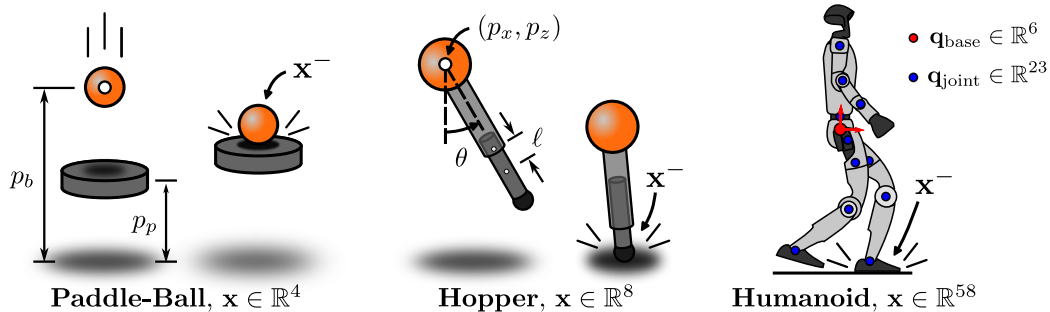


Figure 3: The three systems of study, their configurations, and their impacts.

Thus, stability in the latent representation of the invariant manifold translates to stability in the full-order representation of the invariant manifold. Practically—as we demonstrate below—stability on the invariant manifold can be concluded by exhibiting an asymptotic Lyapunov function for $g : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_z}$; such a Lyapunov function automatically restricts to an asymptotic Lyapunov function on $\mathbf{E}(\mathcal{M})$, which provides the conditions for Theorem 5.

To make a stronger conclusion regarding stability for the full-order dynamics $f : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ on a neighborhood of \mathbf{x}^* in \mathbb{R}^{n_x} , rather than stability on a neighborhood in \mathcal{M} , one may appeal to the literature on attractive invariant manifolds Wiggins (1994). Informally, stability on the invariant manifold plus sufficiently fast convergence normal to the invariant manifold implies stability of the full system. Within the scope of this work, such convergence remains an assumption.

4. Methods

We recall that the stability of a hybrid system can be characterized by its impact-to-impact dynamics. We consider a closed-loop hybrid system with state \mathbf{x} and feedback controller $\mathbf{k}(\mathbf{x})$, learned through reinforcement learning (RL). To study stability of this system, we extract sequences of pre-impact states,

$$\mathfrak{D} = \{ \{ \mathbf{x}_k \}_{k=0}^K : \mathbf{x}_0 \in \mathcal{X}_0 \subset \mathbb{R}^{n_x}, \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \} \quad (7)$$

where $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ is the Poincaré map of the closed-loop system and each $\mathbf{x}_k := \mathbf{x}^-$ is its state just before an impact. The configurations and pre-impact states of the systems considered here—a paddle-ball, a planar hopper, and a Unitree G1 humanoid robot—are illustrated in Figure 3. Using the dataset \mathfrak{D} , we train an autoencoder-based ROM that captures the Poincaré dynamics in a low-dimensional latent space. We then use the learned latent dynamics to predict trajectories and estimate regions of attraction of the FOM. The remainder of this section details each component of this pipeline¹.

Reinforcement Learning Controllers: As described above, the feedback controller $\mathbf{k}(\mathbf{x})$ is an RL policy $\pi_\xi(\mathbf{o})$, where the observation \mathbf{o} is a function of the state for the paddle-ball and hopper, and a function of the state together with auxiliary variables for the humanoid. The paddle-ball policy is trained to regulate the ball at a desired height, whereas the hopper and humanoid policies are trained to maintain a constant forward velocity.

We train controllers for the paddle-ball and hopper using the actor-critic PPO algorithm implemented in Brax Freeman et al. (2021). The learned policy directly outputs bounded torque com-

1. Full code is available at: github.com/sesteban951/halo-latent-locomotion.

mands at 50 Hz. We train a locomotion controller for the G1 humanoid using the actor-critic PPO algorithm implemented in mjlab [Zakka et al. \(2026\)](#) with the `rsl_rl` library. The learned policy outputs joint position setpoints at 50 Hz, which are tracked by joint-level PD controllers to produce motor torques. Additional implementation details are provided in Appendix C.1.

Poincaré Data Collection: To collect training data, we simulate the closed-loop system in MuJoCo MJX [Todorov et al. \(2012\)](#) for a fixed time horizon, using the JAX backend for the paddle-ball and hopper systems and the Warp backend for the G1. At each impact, we use MuJoCo sensor measurements to record the generalized position and velocity in a contact-relative frame—in the world frame for the paddle-ball, relative to the foot position for the hopper, and in a yaw-aligned foot frame for the humanoid. After pruning faulty Poincaré data, such as impacts with sliding contact or contact chatter, we obtain clean sequences of Poincaré returns.

Remark 7 *In practice, impacts can be estimated without dedicated contact sensors. Kinematic methods use foot height and velocity; dynamic methods use changes in generalized velocities, accelerations, or momentum; force methods use joint torques and ground reaction force estimates.*

Autoencoder Architecture: We parameterize the autoencoder using three networks: an *encoder* that projects the FOM state onto a lower-dimensional latent state, $\mathbf{z}_k = \mathbf{E}_\phi(\mathbf{x}_k)$, a *decoder* that reconstructs the FOM state from the latent state, $\hat{\mathbf{x}}_k = \mathbf{D}_\psi(\mathbf{z}_k)$, and a *latent dynamics* network that propagates the latent state forward via learned residual ROM dynamics,

$$\mathbf{z}_{k+1} = \mathbf{g}_\rho(\mathbf{z}_k) := \mathbf{z}_k + \bar{\mathbf{g}}_\rho(\mathbf{z}_k). \quad (8)$$

The encoder and decoder are mirrored MLPs with swish activations, sized $[64, 32, 16]$ and $[16, 32, 64]$ for the hopper and paddle-ball systems, and $[256, 128, 64]$ and $[64, 128, 256]$ for the humanoid. The latent dynamics network is an MLP with swish activations, sized $[64, 64, 64]$ for the hopper and paddle-ball systems, and $[128, 128, 128]$ for the humanoid. The latent dimension n_z is 2 for the paddle-ball, 4 for the hopper, and 12 for the humanoid. More details are provided in Section C.2.

Loss Function: Using the insights from Section 3, we seek parameters ϕ , ψ , and ρ such that the encoder \mathbf{E}_ϕ , decoder \mathbf{D}_ψ , and latent dynamics model \mathbf{g}_ρ approximate the FOM Poincaré map:

$$\inf_{\phi, \psi, \rho} \sum_{\mathbf{x}_k \in \mathcal{D}} \|\mathbf{f}(\mathbf{x}_k) - (\mathbf{D}_\psi \circ \mathbf{g}_\rho \circ \mathbf{E}_\phi)(\mathbf{x}_k)\|^2. \quad (9)$$

The encoder, decoder, and latent dynamics networks are trained jointly using a weighted sum of losses, each enforcing a different aspect of reconstruction accuracy and dynamical consistency. Inspired by [Lutkus et al. \(2025\)](#), we organize these losses into reconstruction, conjugacy, prediction, whitening, and regularization terms. The loss functions are summarized in Table 1 and illustrated in Figure 4.

The reconstruction losses $L_{\text{rec},x}$ and $L_{\text{rec},z}$ ensure that the encoder-decoder pair $(\mathbf{E}_\phi, \mathbf{D}_\psi)$ accurately reconstructs states in both the full-order

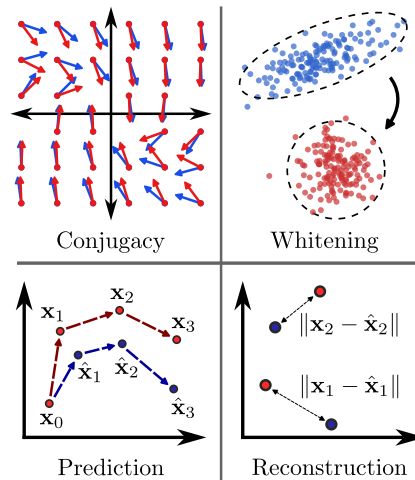


Figure 4: A visual guide of the losses.

Loss	Definition	Purpose
$L_{\text{rec},x}(\phi, \psi)$	$\frac{\lambda_{\text{rec},x}}{BK} \sum_{b,k=1}^{B,K} \left\ \mathbf{x}_k^{(b)} - (\mathbf{D}_\psi \circ \mathbf{E}_\phi)(\mathbf{x}_k^{(b)}) \right\ ^2$	Reconstruction
$L_{\text{rec},z}(\phi, \psi)$	$\frac{\lambda_{\text{rec},z}}{BK} \sum_{b,k=1}^{B,K} \left\ \mathbf{E}_\phi(\mathbf{x}_k^{(b)}) - (\mathbf{E}_\phi \circ \mathbf{D}_\psi \circ \mathbf{E}_\phi)(\mathbf{x}_k^{(b)}) \right\ ^2$	Reconstruction
$L_{\text{fwd}}(\phi, \rho)$	$\frac{\lambda_{\text{fwd}}}{B(K-1)} \sum_{b,k=1}^{B,K-1} \left\ \mathbf{E}_\phi(\mathbf{x}_{k+1}^{(b)}) - (\mathbf{g}_\rho \circ \mathbf{E}_\phi)(\mathbf{x}_k^{(b)}) \right\ ^2$	Conjugacy
$L_{\text{bck}}(\phi, \rho, \psi)$	$\frac{\lambda_{\text{bck}}}{B(K-1)} \sum_{b,k=1}^{B,K-1} \left\ \mathbf{x}_{k+1}^{(b)} - (\mathbf{D}_\psi \circ \mathbf{g}_\rho \circ \mathbf{E}_\phi)(\mathbf{x}_k^{(b)}) \right\ ^2$	Conjugacy
$L_{\text{pred}}(\phi, \rho, \psi)$	$\frac{\lambda_{\text{pred}}}{B(K-1)} \sum_{b=1, k=2}^{B,K} \left\ \mathbf{x}_k^{(b)} - (\mathbf{D}_\psi \circ \mathbf{g}_\rho^{(k-1)} \circ \mathbf{E}_\phi)(\mathbf{x}_1^{(b)}) \right\ ^2$	Prediction
$L_{\text{iso}}(\phi)$	$\lambda_{\text{iso}} \left\ \mathbf{I} - \frac{1}{BK} \sum_{b,k=1}^{B,K} (\mathbf{E}_\phi(\mathbf{x}_k^{(b)}) - \boldsymbol{\mu})(\mathbf{E}_\phi(\mathbf{x}_k^{(b)}) - \boldsymbol{\mu})^\top \right\ _F^2$	Whitening
$L_{\text{reg}}(\phi, \rho, \psi)$	$\lambda_{\text{reg}} \sum_{w_i \in \mathbf{W}(\phi, \psi, \rho)} \ w_i\ ^2$	L2 Regularization

Table 1: Summary of the loss terms. \mathbf{W} denotes trainable weights (excluding biases), and $\mathbf{g}_\rho^{(k-1)}$ denotes repeated application of \mathbf{g}_ρ . In addition, B denotes the mini-batch size and K denotes the trajectory length.

and latent spaces, respectively. The forward and backward conjugacy losses, L_{fwd} and L_{bck} , encourage the learned latent dynamics to align with the FOM Poincaré dynamics. The prediction loss L_{pred} improves multi-step rollout accuracy, encouraging the learned ROM to capture long-horizon trajectory evolution rather than myopic one-step behavior. The whitening loss L_{iso} helps prevent latent space collapse by encouraging the latent covariance to remain close to identity. Finally, the weight regularization loss L_{reg} penalizes large network weights to reduce overfitting to data. The total loss function is given by:

$$L = L_{\text{rec},x} + L_{\text{rec},z} + L_{\text{fwd}} + L_{\text{bck}} + L_{\text{pred}} + L_{\text{iso}} + L_{\text{reg}}. \quad (10)$$

All loss terms are scaled as shown in Table 1, where all weights are set to 1.0 except for the L2 regularization term, which uses $\lambda_{\text{reg}} = 10^{-6}$.

Training: We train autoencoders using the Adam optimizer [Kingma and Ba \(2014\)](#) with shuffled mini-batches of trajectory data. The model is implemented in JAX and Flax [Bradbury et al. \(2018\)](#); [Heek et al. \(2024\)](#). Before training, each state feature is normalized independently using training-set statistics aggregated over all trajectories and time steps. The dataset of trajectories is partitioned into disjoint training, validation, and test sets. More details are provided in Appendix C.2.

Latent Region of Attraction: To approximate the region of attraction (ROA) for the FOM impact-to-impact dynamics, we linearize the latent Poincaré map around the latent equilibrium point² \mathbf{z}^* , yielding the discrete-time linear system $\mathbf{z}_{k+1} = \mathbf{g}_\rho(\mathbf{z}_k) \approx \mathbf{Q}(\mathbf{z}_k - \mathbf{z}^*) + \mathbf{g}_\rho(\mathbf{z}^*)$. Without loss of generality, we shift coordinates so that $\mathbf{z}^* = \mathbf{0}$. We then construct a discrete-time Lyapunov function $V_{\mathbf{z}}(\mathbf{z}) = \mathbf{z}^\top \mathbf{P} \mathbf{z}$ where $\mathbf{P} \succ \mathbf{0}$ satisfies the discrete-time Lyapunov equation $\mathbf{Q}^\top \mathbf{P} \mathbf{Q} - \mathbf{P} + \mathbf{I} = \mathbf{0}$. Next, we construct an estimated ROA around the latent equilibrium and map it through the decoder to obtain an approximation of the full-order ROA. Consider the following set:

$$\mathcal{D} = \{ \mathbf{z} \in \mathbb{R}^{n_z} : \Delta V_{\mathbf{z}}(\mathbf{z}) \leq 0 \} = \left\{ \mathbf{z} \in \mathbb{R}^{n_z} : -\mathbf{z}^\top \mathbf{z} + \mathbf{g}_\rho(\mathbf{z})^\top \mathbf{P} \mathbf{g}_\rho(\mathbf{z}) - \mathbf{z}^\top \mathbf{Q}^\top \mathbf{P} \mathbf{Q} \mathbf{z} \leq 0 \right\}. \quad (11)$$

2. Linearization of $\mathbf{g}_\rho(\mathbf{z})$ is obtained via automatic differentiation using `jax.jac_fwd`, giving $\mathbf{Q} = \frac{\partial \mathbf{g}_\rho}{\partial \mathbf{z}}|_{\mathbf{z}^*}$.

This is the set of points for which the latent candidate Lyapunov function $V_{\mathbf{z}}$ does not increase under the nonlinear latent dynamics. To obtain the largest Lyapunov sublevel set contained in \mathcal{D} , we solve:

$$c^* = \inf_{\mathbf{z} \in \partial\mathcal{D}} \mathbf{z}^\top \mathbf{P} \mathbf{z} \quad (12)$$

using a Monte Carlo sampling strategy³. The sublevel set $\Omega_{c^*} := \{\mathbf{z} : \mathbf{z}^\top \mathbf{P} \mathbf{z} \leq c^*\}$ is taken as the latent ROA estimate. We then sample points in Ω_{c^*} , decode them through \mathbf{D}_ψ to obtain full-order states, and simulate these initial conditions under the FOM dynamics to assess if they remain stable.

5. Results

To validate the capabilities of our autoencoder, we test both its prediction and stability estimation capabilities. To test the encoding and decoding accuracy of the autoencoder, we gather normalized Poincaré data of N full-order rollouts, each of length K for each of our systems. We send this data through the encoder followed by the decoder and calculate

$$\ell_{\text{rec}} = \|\mathbf{x}_k - \mathbf{D}_\psi \circ \mathbf{E}_\phi(\mathbf{x}_k)\|. \quad (13)$$

For each state in the trajectory, we take the mean and standard deviation of ℓ_{rec} at each Poincaré step k . Second, to test the accuracy of the latent dynamics, we take the initial states \mathbf{x}_0 for each of our rollouts, encode the state, perform K steps of the latent dynamics, then decode the latent trajectory, calculating the reconstruction loss as

$$\ell_{\text{dyn}} = \|\mathbf{x}_k - \mathbf{D}_\psi \circ \mathbf{g}_\rho^{(k)} \circ \mathbf{E}_\phi(\mathbf{x}_0)\|. \quad (14)$$

We also take the mean and standard deviation over the Poincaré steps. We observe good performance of the reconstruction and latent dynamics, improving over the rollout as the system stabilizes. We notice that error propagation is minimal despite the length of the rollout, which we attribute to our loss choice penalizing discrepancy in long-horizon prediction, combined with the stability of the systems.

To validate our ROA estimate, we perform the procedure described in Sec. 4: we generate a Lyapunov-based ROA estimate by solving (12), sampling 2,000 points in Ω_{c^*} , and mapping them back to the full-order state space. For comparison, we compute an approximation of the ROA using a naive sampling-based method. Here, we grid the latent space with 4 points per dimension in n_z dimensions, gridding a hypercube of side length equal to the major axis length of the Lyapunov region, decoding these points using the decoder. Next, we roll out all of these samples in parallel simulation, recording those that are stable, characterized by $p_z > p_z^{\text{thresh}}$ not falling below a prescribed threshold over the course of the 150-step trajectory. We observe that the initial conditions in

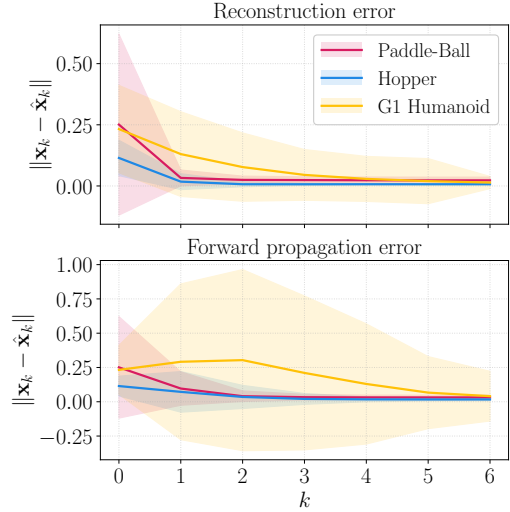


Figure 5: Per-step normalized error of the learned ROM over 3,000 test trajectories for each system. (Top) shows single-step reconstruction error, and (bottom) shows multi-step latent forward-propagation error. Solid curves denote the mean and shaded bands denote $\pm\sigma$.

3. Direct optimization of c^* is challenging as $\partial\mathcal{D}$ is non-convex and implicitly determined by a neural network. We thus use a simple but effective sampling-based sweep to obtain a conservative ROA estimate.

the Lyapunov-based latent ROA consistently have a near-perfect $99.9 \pm 0.1\%$ stability rate, while points from the naive hypercube sampling method have a consistently lower stability rate, for instance $75.6 \pm 10.1\%$ on the G1. In Figures 6 and 7, we plot the stable points from each of these categories. We find that, while the Lyapunov estimate is more conservative, it is far more accurate than the naive sampling-based estimate of the ROA. In Figure 8, we demonstrate the prediction performance of the ROM relative to real-world Unitree G1 experiments.

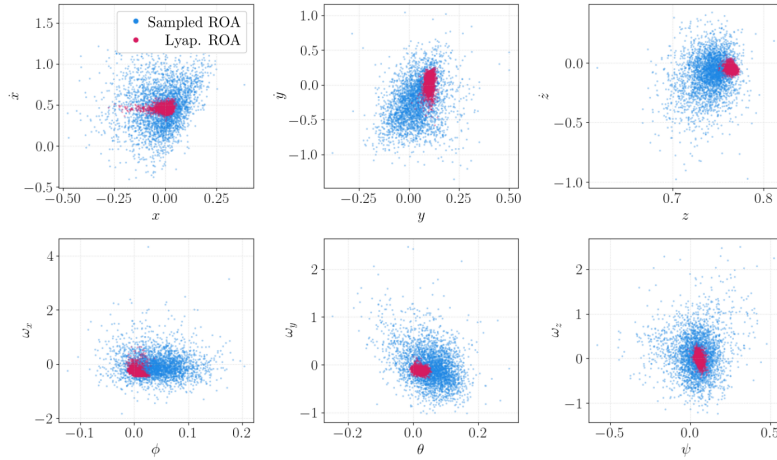


Figure 6: Decoded visualization of the boundary of Ω_z in full-order coordinates of the G1. Plotted states correspond to the center-of-mass position relative to the stance foot.

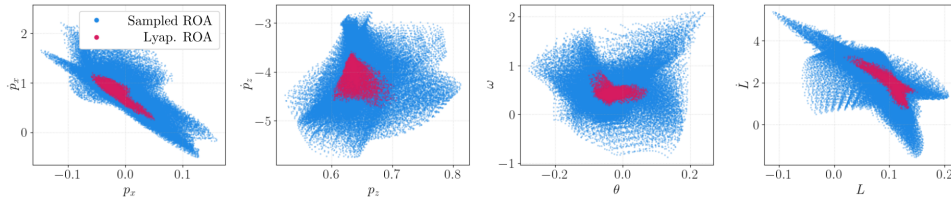


Figure 7: Decoded visualization of the boundary of Ω_z in full-order coordinates of the hopper. Plotted states correspond to the center-of-mass position relative to the hopper foot.

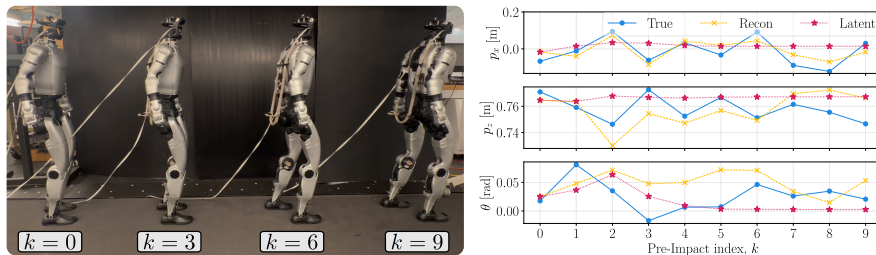


Figure 8: A 9-step rollout with true hardware, encoded-decoded, and latent ROM trajectories.

Conclusion: In this work, we proposed a method of generating reduced-order models using autoencoders to represent and study the stability of hybrid systems. Our technique enables accurate reconstruction of the original dynamical system from a lower-dimensional latent space, as well as estimation of the region of attraction of the original system. This method was validated on three systems of increasing complexity: a paddle-ball, a hopper, and a Unitree G1 humanoid robot.

Acknowledgments

We would like to thank Paul Lutkus, Kaiyuan Wang, and Professor Steven Tu for their insights and guidance. This research is supported by the Technology Innovation Institute.

References

- Aaron D Ames, Kevin Galloway, Koushil Sreenath, and Jessy W Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Trans. Autom. Ctrl.*, 59(4), 2014.
- A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.
- A. Astolfi. Model reduction by moment matching for linear and nonlinear systems. *IEEE Transactions on Automatic Control*, 55(10):2321–2336, 2010.
- P. Benner, S. Gugercin, and K. Wilcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 2015.
- Reinhard Blickhan. The spring-mass model for running and hopping. *Journal of biomechanics*, 22(11-12):1217–1227, 1989.
- James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/jax-ml/jax>.
- S. L. Brunton and J. N. Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.
- Giuseppe C Calafiore and Laurent El Ghaoui. *Optimization models*. Cambridge university press, 2014.
- Guillermo A. Castillo, Bowen Weng, Wei Zhang, and Ayonga Hereid. Data-driven latent space representation for robust bipedal locomotion learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1172–1178, 2024. doi: 10.1109/ICRA57147.2024.10610978.
- Yu-Ming Chen, Hien Bui, and Michael Posa. Reinforcement learning for reduced-order models of legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5801–5807, 2024a. doi: 10.1109/ICRA57147.2024.10610747.
- Yu-Ming Chen, Jianshu Hu, and Michael Posa. Beyond inverted pendulums: Task-optimal simple models of legged locomotion. *IEEE Transactions on Robotics*, 40:2582–2601, 2024b. doi: 10.1109/TRO.2024.3386390.
- Geir E Dullerud and Fernando Paganini. *A course in robust control theory: a convex approach*, volume 36. Springer Science & Business Media, 2013.

- Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.
- C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - a differentiable physics engine for large scale rigid body simulation, 2021. URL <http://github.com/google/brax>.
- A. Girard and G. J. Pappas. Hierarchical control system design using approximate simulation. *Automatica*, 45(2):566–571, 2009.
- G. Haller and P. Ponsioen. Nonlinear normal modes and spectral submanifolds: existence, uniqueness and use in model reduction. *Nonlinear Dynamics*, 86(3):1493–1534, 2016.
- Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024. URL <http://github.com/google/flax>.
- Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No. 01CH37180)*, volume 1, pages 239–246. IEEE, 2001.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Matthew D. Kvalheim and Eduardo D. Sontag. Autoencoding dynamics: Topological limitations and capabilities, 2025. URL <https://arxiv.org/abs/2511.04807>.
- John M Lee. *Introduction to Smooth Manifolds*. Springer, 2013.
- Chenhao Li, Elijah Stanger-Jones, Steve Heim, and Sangbae Kim. FLD: Fourier latent dynamics for structured motion representation and learning. *arXiv preprint arXiv:2402.13820*, 2024.
- Paul Lutkus, Kaiyuan Wang, Lars Lindemann, and Stephen Tu. Latent representations for control design with provable stability and safety guarantees. *arXiv preprint arXiv:2505.23210*, 2025.
- K. Nakamura, L. Peters, and A. Bajcsy. Generalizing safety beyond collision-avoidance via latent-space reachability analysis. In *Robotics: Science and Systems*, 2025.
- Marc H Raibert. *Legged robots that balance*. MIT press, 1986.
- Sebastian Starke, Ian Mason, and Taku Komura. DeepPhase: periodic autoencoders for learning motion phase manifolds. *ACM Trans. Graph.*, 41(4), July 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530178. URL <https://doi.org/10.1145/3528223.3530178>.
- P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.
- Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

P. Wensing and S. Revzen. Template models for control. *Bioinspired Legged Locomotion*, pages 240–266, 2017.

Eric R Westervelt, Jessy W Grizzle, and Daniel E Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(1):42–56, 2003.

Eric R Westervelt, Jessy W Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.

Stephen Wiggins. *Normally hyperbolic invariant manifolds in dynamical systems*, volume 105. Springer Science & Business Media, 1994.

Kevin Zakka, Qiayuan Liao, Brent Yi, Louis Le Lay, Koushil Sreenath, and Pieter Abbeel. mjlab: A lightweight framework for gpu-accelerated robot learning, 2026. URL <https://arxiv.org/abs/2601.22074>.

Appendix A. Full-Order Model Hybrid Dynamics

Legged robots that make and break contact with the environment can be modeled as nonlinear dynamical systems with impulse effects, which in this work is referred to as the *full-order model*. The generalized configuration for these systems is $\mathbf{q} \in \mathcal{Q} \subseteq \mathbb{R}^n$ with state $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{T}\mathcal{Q} \subseteq \mathbb{R}^{2n} = \mathbb{R}^{n_x}$. Using Euler-Lagrange equations, we define the continuous dynamics:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\mathbf{u} + \mathbf{J}_c^\top(\mathbf{q})\boldsymbol{\lambda}, \quad (15)$$

$$\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{0} \quad (16)$$

where $\mathbf{M} : \mathcal{Q} \rightarrow \mathbb{R}^{n \times n}$ is the positive definite mass-inertia matrix, $\mathbf{H} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^n$ contains centrifugal, Coriolis, and gravitational terms, $\mathbf{B} \in \mathbb{R}^{n \times n_u}$ is the actuation matrix, and $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the control input. Additionally, $\mathbf{J} : \mathcal{Q} \rightarrow \mathbb{R}^{n_c \times n}$ is the contact Jacobian of a *holonomic constraint* describing contact of the robot with the environment with corresponding constraint wrench $\boldsymbol{\lambda} \in \mathbb{R}^{n_c}$. Furthermore, (15) and (16) can be combined to write the state-space dynamics representation:

$$\dot{\mathbf{x}} = \mathbf{f}_c(\mathbf{x}, \mathbf{u}) := \begin{bmatrix} \dot{\mathbf{q}} \\ -\mathbf{M}^{-1}(\mathbf{q})\tilde{\mathbf{H}}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}(\mathbf{q})\tilde{\mathbf{B}}(\mathbf{q}) \end{bmatrix} \mathbf{u}, \quad (17)$$

where the dynamics $\mathbf{f}_c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ are continuously differentiable.

Discrete jumps in hybrid systems occur when a foot or hand impacts with the environment. A *switching surface* is designed to capture such events. Given a continuously differentiable function $s : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$, let $\mathcal{S} \subset \mathbb{R}^{n_x}$ denote the switching surface as:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid s(\mathbf{x}) = 0, \dot{s}(\mathbf{x}) < 0\}. \quad (18)$$

The dynamics undergo discrete transitions which are represented by:

$$\mathbf{x}^+ = \boldsymbol{\Delta}(\mathbf{x}^-) \quad (19)$$

where $\boldsymbol{\Delta} : \mathcal{S} \rightarrow \mathbb{R}^{n_x}$ is the continuously differentiable *reset map*, $\mathbf{x}^- \in \mathcal{S}$ denotes the *pre-impact state* just before impact and $\mathbf{x}^+ \in \mathbb{R}^{n_x} \setminus \mathcal{S}$ denotes the *post-impact state* just after impact. Combining the continuous dynamics (17) and the reset map (19), the full-order dynamics can be written as a hybrid control system:

$$\mathcal{HC} = \begin{cases} \dot{\mathbf{x}} = \mathbf{f}_c(\mathbf{x}, \mathbf{u}) & \text{if } \mathbf{x} \in \mathbb{R}^{n_x} \setminus \mathcal{S} \\ \mathbf{x}^+ = \boldsymbol{\Delta}(\mathbf{x}^-) & \text{if } \mathbf{x}^- \in \mathcal{S}. \end{cases} \quad (20)$$

Furthermore, given a locally Lipschitz feedback controller $\mathbf{k} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ we obtain the closed-loop hybrid system:

$$\mathcal{H} = \begin{cases} \dot{\mathbf{x}} = \mathbf{f}_c(\mathbf{x}, \mathbf{k}(\mathbf{x})) & \text{if } \mathbf{x} \in \mathbb{R}^{n_x} \setminus \mathcal{S} \\ \mathbf{x}^+ = \boldsymbol{\Delta}(\mathbf{x}^-) & \text{if } \mathbf{x}^- \in \mathcal{S}. \end{cases} \quad (21)$$

Let the flow of these dynamics be denoted by $\varphi_t(\mathbf{x})$, which returns the state reached at time t under the continuous dynamics (17) when starting from state $\mathbf{x} \in \mathcal{X}$. To further characterize the discrete evolution of (21), define the *time-to-impact function* $T_I : \tilde{\mathcal{S}} \rightarrow \mathbb{R}_{>0}$ as:

$$T_I(\mathbf{x}) := \inf\{t \geq 0 \mid \varphi_t(\boldsymbol{\Delta}(\mathbf{x})) \in \mathcal{S}\}, \quad (22)$$

where $\tilde{\mathcal{S}} := \{\mathbf{x} \in \mathcal{S} \mid T_I(\mathbf{x}) \in (0, \infty)\}$, which captures the time passed between discrete transitions. The above formulation allows for casting the hybrid system (21) as a discrete-time system via the *Poincaré map*:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k), \quad \mathbf{f}(\mathbf{x}) := \varphi_{T_I(\mathbf{x})}(\Delta(\mathbf{x})), \quad \mathbf{f} : \tilde{\mathcal{S}} \rightarrow \mathcal{S}. \quad (23)$$

In the literature, for legged robots, the Poincaré map $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$ is often referred to as the *step-to-step dynamics* of the FOM. With an abuse of notation, denote the Poincaré map as $\mathbf{f} : \mathcal{S} \rightarrow \mathcal{S}$ with the understanding that \mathbf{f} is a partial function (see Ames et al. (2014); Westervelt et al. (2003)). The Poincaré map provides a powerful tool for studying the stability of orbits of (21). The flow of (21) is said to be periodic with period $T \geq 0$ if there exists a state $\mathbf{x}^* \in \mathcal{S}$ such that $\varphi_T(\Delta(\mathbf{x}^*)) = \mathbf{x}^*$. This induces a *periodic orbit*:

$$\mathcal{O} := \{\varphi_t(\Delta(\mathbf{x}^*)) \mid t \in [0, T], T = T_I(\mathbf{x}^*)\}. \quad (24)$$

Importantly, stability of \mathcal{O} is equivalent to that of \mathbf{f} as shown in the following theorem.

Theorem 8 (Westervelt et al. (2018)) *Let the closed loop dynamics \mathbf{f} and reset map Δ be continuously differentiable and $\Delta(\mathcal{S}) \cap \mathcal{S} = \emptyset$. Then, the following statements hold:*

- $\mathbf{x}^* \in \mathcal{S}$ is a stable equilibrium point for $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \iff$ the orbit $\mathcal{O}(\Delta(\mathbf{x}^*))$ is stable.
- $\mathbf{x}^* \in \mathcal{S}$ is an asymptotically stable equilibrium point for $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \iff$ the orbit $\mathcal{O}(\Delta(\mathbf{x}^*))$ is asymptotically stable.
- $\mathbf{x}^* \in \mathcal{S}$ is an exponentially stable equilibrium point for $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \iff$ the orbit $\mathcal{O}(\Delta(\mathbf{x}^*))$ is exponentially stable.

Thus, establishing stability of the Poincaré map (23) certifies stability of the full hybrid system (21).

Appendix B. Proofs

In this section, we provide proofs of the theoretical results in the paper. For convenience, we restate all results below.

B.1. Proof of Lemma 1

Lemma *Let $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ and $n_z \in (0, \text{rank}(\mathbf{A})]_{\mathbb{Z}}$. Consider:*

$$\inf_{\mathbf{E} \in \mathbb{R}^{n_z \times n_x}, \mathbf{D} \in \mathbb{R}^{n_x \times n_z}, \mathbf{Q} \in \mathbb{R}^{n_z \times n_z}} \sup_{\mathbf{x}_0 \in \mathbb{R}^{n_x}, \|\mathbf{x}_0\|=1} \|\mathbf{A}\mathbf{x}_0 - \mathbf{D}\mathbf{Q}\mathbf{E}\mathbf{x}_0\|^2. \quad (25)$$

If $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ is an SVD of \mathbf{A} , then $\mathbf{D} = \mathbf{U}_{n_z}$, $\mathbf{Q} = \Sigma_{n_z}$, $\mathbf{E} = \mathbf{V}_{n_z}^\top$, where $\mathbf{U}_{n_z}, \mathbf{V}_{n_z}$ are the first n_z columns of \mathbf{U}, \mathbf{V} and $\Sigma_{n_z} = \text{diag}(\sigma_1, \dots, \sigma_{n_z})$, are optimal solutions.

Proof The maximization, $\sup_{\mathbf{x}_0 \in \mathbb{R}^{n_x}, \|\mathbf{x}_0\|=1} \|\mathbf{A}\mathbf{x}_0 - \mathbf{D}\mathbf{Q}\mathbf{E}\mathbf{x}_0\|^2$ (for $\|\cdot\|$ the ℓ_2 -norm) is simply the square of the induced matrix 2-norm, $\|\mathbf{A} - \mathbf{D}\mathbf{Q}\mathbf{E}\|_2^2$. Hence, the problem reduces to

$$\inf_{\mathbf{E}, \mathbf{D}, \mathbf{Q}} \|\mathbf{A} - \mathbf{D}\mathbf{Q}\mathbf{E}\|_2^2. \quad (26)$$

Since $\text{rank}(\mathbf{DQE}) \leq n_z$ for every pair of $\mathbf{D}, \mathbf{Q}, \mathbf{E}$, it follows that

$$\inf_{\mathbf{E}, \mathbf{D}, \mathbf{Q}} \|\mathbf{A} - \mathbf{DQE}\|_2^2 \geq \inf_{\mathbf{M} \in \mathbb{R}^{n_x \times n_x}} \|\mathbf{A} - \mathbf{M}\|_2^2 \text{ s.t. } \text{rank}(\mathbf{M}) \leq n_z. \quad (27)$$

The lower bound is a well-known rank-constrained minimization (Calafiore and El Ghaoui, 2014, p. 134). For an SVD $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, the Eckart-Young theorem tells us that an optimal solution \mathbf{M} is the n_z -truncation of the SVD, $\mathbf{M} = \mathbf{U}_{n_z}\mathbf{\Sigma}_{n_z}\mathbf{V}_{n_z}^\top$. Here, $\mathbf{U}_{n_z}, \mathbf{V}_{n_z}$ are the first n_z columns of \mathbf{U}, \mathbf{V} , respectively, and $\mathbf{\Sigma}_{n_z} = \text{diag}(\sigma_1, \dots, \sigma_{n_z})$. Since $\mathbf{D} = \mathbf{U}_{n_z}, \mathbf{Q} = \mathbf{\Sigma}_{n_z}, \mathbf{E} = \mathbf{V}_{n_z}^\top$ are feasible for the original problem, the lower bound above implies their optimality. ■

B.2. Proof of Lemma 2

Lemma *Suppose $\mathbf{A} = \text{diag}(\lambda_1, \dots, \lambda_{n_x})$, with $|\lambda_1| \geq \dots \geq |\lambda_{n_x}|$. Let $n_z \in (0, \text{rank}(\mathbf{A}))_{\mathbb{Z}}$. If the unstable and center subspaces of \mathbf{A} together have dimension $\leq n_z$, then Problem (5) is feasible and an optimal solution is given by the n_z -truncation:*

$$\hat{\mathbf{D}} = \begin{bmatrix} I_{n_z} \\ 0_{(n_x-n_z) \times n_z} \end{bmatrix}, \quad \hat{\mathbf{Q}} = \text{diag}(\lambda_1, \dots, \lambda_{n_z}), \quad \hat{\mathbf{E}} = [I_{n_z} \quad 0_{n_z \times (n_x-n_z)}]. \quad (28)$$

Proof We begin by considering the simpler one-step problem $\inf_{\mathbf{E}, \mathbf{D}, \mathbf{Q}} \|\mathbf{A}^k - \mathbf{DQ}^k\mathbf{E}\|_F^2$. Since $\text{rank}(\mathbf{DQ}^k\mathbf{E}) \leq n_z$, it must be that for each fixed $k \geq 0$,

$$\inf_{\mathbf{E}, \mathbf{D}, \mathbf{Q}} \|\mathbf{A}^k - \mathbf{DQ}^k\mathbf{E}\|_F^2 \geq \inf_{\mathbf{M}} \|\mathbf{A}^k - \mathbf{M}\|_F^2, \text{ s.t. } \text{rank}(\mathbf{M}) \leq n_z. \quad (29)$$

Now, fix $k \geq 0$. Since \mathbf{A} is diagonal, $\mathbf{A}^k = \text{diag}(\lambda_1^k, \dots, \lambda_{n_x}^k)$. As such, any off-diagonal entry \mathbf{M}_{ij} raises the Frobenius norm error. This implies that we can further reduce the problem to:

$$\left(\inf_{\mathbf{M}} \|\mathbf{A}^k - \mathbf{M}\|_F^2 \right)_{\text{s.t. } \text{rank}(\mathbf{M}) \leq n_z} = \left(\inf_{\mathbf{M}} \|\mathbf{A}^k - \mathbf{M}\|_F^2 \right)_{\text{s.t. } \text{rank}(\mathbf{M}) \leq n_z, \mathbf{M} \text{ diag.}} \quad (30)$$

Since \mathbf{M} is constrained to have $\text{rank} \leq n_z$, the problem amounts to choosing at most n_z nonzero diagonal entries of \mathbf{M} . Since the entries of \mathbf{A} are ordered by decreasing magnitude an optimal \mathbf{M} is $\mathbf{M} = \text{diag}(\lambda_1^k, \dots, \lambda_{n_z}^k, 0_{n_x-n_z})$. Since $\mathbf{M} = \hat{\mathbf{D}}\hat{\mathbf{Q}}^k\hat{\mathbf{E}}$, for $\hat{\mathbf{D}}, \hat{\mathbf{Q}}, \hat{\mathbf{E}}$ as in the statement of the lemma,

$$\inf_{\mathbf{E}, \mathbf{D}, \mathbf{Q}} \|\mathbf{A}^k - \mathbf{DQ}^k\mathbf{E}\|_F^2 \geq \|\mathbf{A}^k - \hat{\mathbf{D}}\hat{\mathbf{Q}}^k\hat{\mathbf{E}}\|_F^2, \quad \forall k \geq 0. \quad (31)$$

If the unstable and center subspaces of \mathbf{A} together have dimension $\leq n_z$, for each $j > n_z, |\lambda_j| < 1$. Hence, one has

$$\sum_{k=0}^{\infty} \|\mathbf{A}^k - \hat{\mathbf{D}}\hat{\mathbf{Q}}^k\hat{\mathbf{E}}\|_F^2 = \sum_{k=0}^{\infty} \sum_{j=n_z+1}^{n_x} |\lambda_j^k|^2 = \sum_{j=n_z+1}^{n_x} \sum_{k=0}^{\infty} |\lambda_j^k|^2 < \infty, \quad (32)$$

which confirms feasibility of the proposed solution. Applying (31) for each k yields

$$\inf_{\mathbf{E}, \mathbf{D}, \mathbf{Q}} \sum_{k=0}^{\infty} \|\mathbf{A}^k - \mathbf{DQ}^k\mathbf{E}\|_F^2 \geq \sum_{k=0}^{\infty} \|\mathbf{A}^k - \hat{\mathbf{D}}\hat{\mathbf{Q}}^k\hat{\mathbf{E}}\|_F^2, \quad (33)$$

which establishes the optimality of the proposed solution. ■

B.3. Proof of Lemma 3

Lemma *Let $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ be a map defining a discrete-time system with smooth invariant manifold $\mathcal{M} \subseteq \mathbb{R}^{n_x}$. For $\dim(\mathcal{M}) = k$:*

1. Perfect encoder-decoder pair on \mathcal{M} : *there exists a smooth encoder $\mathbf{E} : U \subseteq \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$ and a smooth decoder $\mathbf{D} : W \subseteq \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_x}$ satisfying $\mathbf{D}(\mathbf{E}(\mathbf{x})) = \mathbf{x}$ for all $\mathbf{x} \in \mathcal{M}$, where $n_z \leq 2k$ and U, W are open neighborhoods of \mathcal{M} and $\mathbf{E}(\mathcal{M})$ respectively,*
2. Latent system: *there exists a map $\mathbf{g} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_z}$ satisfying $\mathbf{f}(\mathbf{x}) = \mathbf{D} \circ \mathbf{g} \circ \mathbf{E}(\mathbf{x}) \forall \mathbf{x} \in \mathcal{M}$.*

Proof Whitney’s embedding theorem guarantees the existence of a smooth embedding $\bar{\mathbf{E}} : \mathcal{M} \rightarrow \mathbb{R}^{n_z}$, where $n_z \leq 2k$. Since $\bar{\mathbf{E}}$ is an embedding, we may restrict its codomain to obtain a diffeomorphism $\tilde{\mathbf{E}} : \mathcal{M} \rightarrow \bar{\mathbf{E}}(\mathcal{M})$. Since $\tilde{\mathbf{E}}$ is a diffeomorphism, it has a smooth inverse $\tilde{\mathbf{D}} : \bar{\mathbf{E}}(\mathcal{M}) \rightarrow \mathcal{M}$, satisfying $\tilde{\mathbf{D}}(\tilde{\mathbf{E}}(\mathbf{x})) = \tilde{\mathbf{D}}(\bar{\mathbf{E}}(\mathbf{x})) = \mathbf{x}$ for all $\mathbf{x} \in \mathcal{M}$.

Now, we extend the (co)domains of $\bar{\mathbf{E}}$ and $\tilde{\mathbf{D}}$. By (Lee, 2013, Lem. 5.34), there exists a neighborhood $U \subseteq \mathbb{R}^{n_x}$ of \mathcal{M} and a smooth function $\mathbf{E} : U \rightarrow \mathbb{R}^{n_z}$ for which $\mathbf{E}|_{\mathcal{M}} = \bar{\mathbf{E}}$. Since \mathcal{M} is embedded in \mathbb{R}^{n_x} , the inclusion map $\iota_{\mathcal{M}} : \mathcal{M} \hookrightarrow \mathbb{R}^{n_x}$ taking $\mathbf{x} \in \mathcal{M}$ to \mathbf{x} as an element of \mathbb{R}^{n_x} is smooth; therefore the composition $\iota_{\mathcal{M}} \circ \tilde{\mathbf{D}} : \bar{\mathbf{E}}(\mathcal{M}) \rightarrow \mathbb{R}^{n_x}$ is smooth. Applying (Lee, 2013, Lem. 5.34) to $\iota_{\mathcal{M}} \circ \tilde{\mathbf{D}}$, we conclude the existence of a neighborhood $W \subseteq \mathbb{R}^{n_z}$ of $\bar{\mathbf{E}}(\mathcal{M})$ and a map $\mathbf{D} : W \rightarrow \mathbb{R}^{n_x}$ for which $\mathbf{D}|_{\bar{\mathbf{E}}(\mathcal{M})} = \iota_{\mathcal{M}} \circ \tilde{\mathbf{D}}$. By the extension property, it follows that $\mathbf{D}(\mathbf{E}(\mathbf{x})) = \mathbf{x}$ for all $\mathbf{x} \in \mathcal{M}$, which establishes (1).

Now, we construct the dynamical system $\mathbf{z}_{k+1} = \mathbf{g}(\mathbf{z}_k)$, satisfying $\mathbf{f}(\mathbf{x}) = \mathbf{D}(\mathbf{g}(\mathbf{E}(\mathbf{x})))$ for all $\mathbf{x} \in \mathcal{M}$. Define $\tilde{\mathbf{g}} : \mathbf{E}(\mathcal{M}) \rightarrow \mathbb{R}^{n_z}$ by $\mathbf{z} \mapsto \mathbf{E} \circ \mathbf{f} \circ \mathbf{D}(\mathbf{z})$. If $\mathbf{z} \in \mathbf{E}(\mathcal{M})$, then $\mathbf{D}(\mathbf{z}) \in \mathcal{M}$ by part (1), which implies that $\mathbf{E} \circ \mathbf{f} \circ \mathbf{D}(\mathbf{z}) \in \mathbf{E}(\mathcal{M})$. So, $\tilde{\mathbf{g}}(\mathbf{z}) = \mathbf{E} \circ \mathbf{f} \circ \mathbf{D}(\mathbf{z})$ and $\mathbf{E} \circ \mathbf{f} \circ \mathbf{D}(\mathbf{z}) \in \mathbf{E}(\mathcal{M})$ implies that $\mathbf{D} \circ \tilde{\mathbf{g}}(\mathbf{z}) = \mathbf{f} \circ \mathbf{D}(\mathbf{z})$ for all $\mathbf{z} \in \mathbf{E}(\mathcal{M})$. We conclude that $\mathbf{D} \circ \tilde{\mathbf{g}} \circ \mathbf{E}(\mathbf{x}) = \mathbf{f}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{M}$. Extending $\tilde{\mathbf{g}}$ to \mathbb{R}^{n_z} (we don’t ask for regularity in $\mathbb{R}^{n_z} \setminus \mathbf{E}(\mathcal{M})$), the result follows. ■

B.4. Proof of Theorem 5

Theorem *Consider the setting of Lemma 3 and its proof. Let $\tilde{\mathbf{g}} : \mathbf{E}(\mathcal{M}) \rightarrow \mathbf{E}(\mathcal{M})$ be the restriction of \mathbf{g} to $\mathbf{E}(\mathcal{M})$ and $\tilde{\mathbf{f}} : \mathcal{M} \rightarrow \mathcal{M}$ the restriction of \mathbf{f} to \mathcal{M} . Suppose $\mathbf{z}^* \in \mathbf{E}(\mathcal{M})$ is a locally asymptotically stable fixed point of $\tilde{\mathbf{g}}$. Then, $\mathbf{x}^* = \mathbf{D}(\mathbf{z}^*)$ is a locally asymptotically stable fixed point of $\tilde{\mathbf{f}}$. Further, if \mathbf{z} belongs to the region of attraction of \mathbf{z}^* for $\tilde{\mathbf{g}}$, then $\mathbf{D}(\mathbf{z})$ belongs to the region of attraction of \mathbf{x}^* for $\tilde{\mathbf{f}}$.*

Proof In the proof of Lemma 3, we demonstrated that the map $\tilde{\mathbf{E}} : \mathcal{M} \rightarrow \mathbf{E}(\mathcal{M})$ is a diffeomorphism with inverse $\tilde{\mathbf{D}} : \mathbf{E}(\mathcal{M}) \rightarrow \mathcal{M}$. Further, we showed that $\tilde{\mathbf{f}} = \tilde{\mathbf{D}} \circ \tilde{\mathbf{g}} \circ \tilde{\mathbf{E}}$. First, we note that this conjugacy implies $\tilde{\mathbf{f}}^{(j)} = \tilde{\mathbf{D}} \circ \tilde{\mathbf{g}}^{(j)} \circ \tilde{\mathbf{E}}$ for each $j \geq 0$, where (j) represents j compositions (with $j = 0$ being the identity map).

This composition equality implies that $\mathbf{x}^* = \mathbf{D}(\mathbf{z}^*)$ is a fixed point of $\tilde{\mathbf{f}}$. As local asymptotic stability of fixed points is preserved by diffeomorphism, we conclude local asymptotic stability. Now, we consider the domain of attraction. If \mathbf{z} belongs to the region of attraction of \mathbf{z}^* , then $\lim_{j \rightarrow \infty} \tilde{\mathbf{g}}^{(j)}(\mathbf{z}) = \mathbf{z}^*$. From the composition rule above, $\tilde{\mathbf{f}}^{(j)}(\tilde{\mathbf{D}}(\mathbf{z})) = \tilde{\mathbf{D}} \circ \tilde{\mathbf{g}}^{(j)} \circ \tilde{\mathbf{E}} \circ \tilde{\mathbf{D}}(\mathbf{z}) = \tilde{\mathbf{D}} \circ \tilde{\mathbf{g}}^{(j)}(\mathbf{z})$ for each $j \geq 0$. Passing to the limit, $\lim_{j \rightarrow \infty} \tilde{\mathbf{f}}^{(j)}(\tilde{\mathbf{D}}(\mathbf{z})) = \tilde{\mathbf{D}}(\lim_{j \rightarrow \infty} \tilde{\mathbf{g}}^{(j)}(\mathbf{z})) = \tilde{\mathbf{D}}(\mathbf{z}^*) = \mathbf{x}^*$. We conclude that $\tilde{\mathbf{D}}(\mathbf{z}) = \mathbf{D}(\mathbf{z})$ belongs to the region of attraction of \mathbf{x}^* for $\tilde{\mathbf{f}}$. ■

Appendix C. Implementation Details

This section contains implementation details spanning the entire proposed pipeline. The full code implementation can be found here: <https://github.com/sesteban951/halo-latent-locomotion>.

C.1. Reinforcement Learning Parameters

For all three systems, the RL policy serves as the feedback controller used to stabilize the closed-loop dynamics and generate trajectories for subsequent ROM learning. Table 2 summarizes the controller implementation details. Additionally, because the three systems differ substantially in morphology and task structure, their observation spaces are specified separately in Tables 3–5.

Property	Paddle-Ball	Hopper	G1 Humanoid
Framework	Brax	Brax	mjlab
Algorithm	PPO	PPO	PPO
Observation dim.	4	8	80
Action dim.	1	2	23
Action type	Direct force	Direct torque/force	Position setpoints (PD)
Policy network	$[32] \times 4$	$[32] \times 4$	$[512, 256, 128]$
Value network	$[256] \times 5$	$[256] \times 5$	$[512, 256, 128]$
Activation	Swish	Swish	ELU
Control frequency	50 Hz	50 Hz	50 Hz
Number of envs	2048	4096	4096
Training steps	100M	60M	980M
Episode length	10 sec.	12 sec.	20 sec.

Table 2: RL controller implementation details.

Observation	Size	Purpose
Ball position	1	Track ball height for interception
Paddle position	1	Track paddle location
Ball velocity	1	Predict incoming ball motion
Paddle velocity	1	Measure paddle motion
Total	4	

Table 3: Paddle-ball RL observations.

C.2. Autoencoder Parameters

We train each autoencoder using the Adam optimizer. Prior to training, the Poincaré section data is normalized to zero mean and unit variance per feature, ensuring that all state components contribute equally to the loss regardless of their physical units or magnitudes. The remaining hyperparameters, which vary across the three systems to account for differences in state dimension and dataset size, are summarized in Table 6.

Observation	Size	Purpose
Body height	1	Regulate vertical posture
Body orientation	2	Regulate torso orientation
Leg position	1	Measure leg configuration
Horizontal velocity	1	Track forward motion
Vertical velocity	1	Track vertical motion
Angular velocity	1	Measure body rotation rate
Leg velocity	1	Measure leg motion
Total	8	

Table 4: Hopper RL observations.

Observation	Size	Purpose
Angular velocity	3	Measure base rotational motion
Projected gravity	3	Regulate body orientation w.r.t. gravity
Velocity command	3	Specify desired locomotion velocity command
Gait phase	2	Encode periodic gait timing
Joint positions	23	Measure robot configuration
Joint velocities	23	Measure joint motion
Last action	23	Provide action history for smoothing
Total	80	

Table 5: G1 humanoid RL observations.

We normalize each state feature independently using training-set statistics aggregated over all trajectories and time steps, yielding $\mathbf{x} = (\mathbf{x} - \boldsymbol{\mu})/\boldsymbol{\sigma}$. This standardization reduces variation in scale across state components and improves the conditioning of the learning problem. Without this normalization, state components with larger magnitudes would dominate the loss and bias learning away from smaller-scale but still informative dynamics.

During training, the autoencoder operates on these normalized state trajectories. The same feature-wise normalization is applied consistently across the training, validation, and test splits so that the encoder, decoder, and latent dynamics model all learn on a common normalized representation of the FOM state. For evaluation, we reuse the same training-set statistics to scale residuals on a per-feature basis. In addition, for the plotting metric, we normalize by state dimension when computing the per-step RMS error across state components. This yields a normalized, dimensionless error measure that is more comparable across systems and state dimensions.

Property	Paddle-Ball	Hopper	G1 Humanoid
FOM state dim., n_x	4	8	59
ROM state dim., n_z	2	4	12
Encoder layers, \mathbf{E}_ϕ	[64, 32, 16]	[64, 32, 16]	[256, 128, 64]
Decoder layers, \mathbf{D}_ψ	[16, 32, 64]	[16, 32, 64]	[64, 128, 256]
Dynamics layers, \mathbf{g}_ρ	[64] \times 3	[64] \times 3	[128] \times 3
Activation	Swish	Swish	Swish
Training steps	5000	15000	30000
Learning rate	10^{-3}	10^{-3}	10^{-3}
Trajectory length, K	6	8	8
Mini-batch size, B	512	1024	512
Training Trajectories	40960	81920	108403
Testing Trajectories	8192	8192	36230
Validation fraction, %	5%	10%	17%

Table 6: Autoencoder ROM training details. The G1 state vector is 59-dimensional due to the quaternion representation of base orientation ($n_q = 30$, $n_v = 29$), though the underlying state space is 58-dimensional.