

# Conformal Koopman for Embedded Nonlinear Control with Statistical Robustness: Theory and Real-World Validation

Koki Hirano and Hiroyasu Tsukamoto

**Abstract**—We propose a fully data-driven, Koopman-based framework for statistically robust control of discrete-time nonlinear systems with linear embeddings. Establishing a connection between the Koopman operator and contraction theory, it offers distribution-free probabilistic bounds on the state tracking error under Koopman modeling uncertainty. Conformal prediction is employed here to rigorously derive a bound on the state-dependent modeling uncertainty throughout the trajectory, ensuring safety and robustness without assuming a specific error prediction structure or distribution. Unlike prior approaches that merely combine conformal prediction with Koopman-based control in an open-loop setting, our method establishes a closed-loop control architecture with formal guarantees that explicitly account for both forward and inverse modeling errors. Also, by expressing the tracking error bound in terms of the control parameters and the modeling errors, our framework offers a quantitative means to formally enhance the performance of arbitrary Koopman-based control. We validate our method both in numerical simulations with the Dubins car and in real-world experiments with a highly nonlinear flapping-wing drone. The results demonstrate that our method indeed provides formal safety guarantees while maintaining accurate tracking performance under Koopman modeling uncertainty.

## I. INTRODUCTION

Data-driven control with the Koopman operator has shown potential in applying linear control techniques to nonlinear dynamical systems. By modeling the operator with finite-dimensional representations, the original nonlinear system can be lifted into a higher-dimensional latent space where the dynamics evolutions are approximately linear. In the lifted space, well-established linear control methods can be systematically applied to design computationally efficient and interpretable control policies [1], [2].

Recent studies have employed deep neural networks to learn the Koopman operator and its lifting functions [2]–[5]. However, two primary sources of modeling error persist: (i) the learned latent dynamics retain residual nonlinearity, as a finite-dimensional set of observables cannot perfectly linearize arbitrary nonlinear dynamics, and (ii) an encoder–decoder mismatch arises since the decoder is not explicitly constrained to be the inverse of the encoder. Such model approximation errors can critically impact control performance, especially in applications that require formal safety or stability guarantees. Existing Koopman-based control frameworks typically address these issues through assumptions on the

This work benefited from technical discussions within DARPA’s Safe and Assured Foundation Robots for Open eNvironments (SAFRON) Program, under contract number HR0011-25-3-0331. The authors are with the Department of Aerospace Engineering, The Grainger College of Engineering, University of Illinois Urbana-Champaign, Urbana, Illinois 61801, [khirano@illinois.edu](mailto:khirano@illinois.edu), [hiroyasu@illinois.edu](mailto:hiroyasu@illinois.edu).

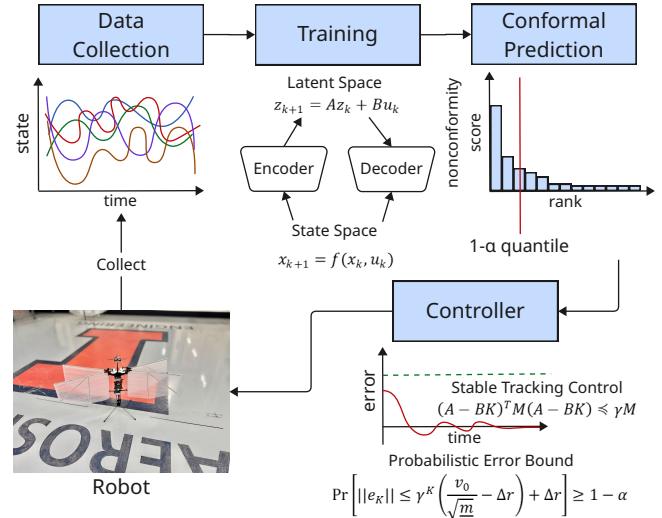


Fig. 1: Overview of the conformal Koopman for embedded nonlinear control with statistical robustness.

modeling error and prediction structure. Some approaches derive Lipschitz bounds from training residuals to design robust model predictive control (MPC) formulations [6], while others incorporate auxiliary controllers or ensemble-based uncertainty modeling to compensate for latent space inaccuracies [7], [8]. These structural assumptions, however, are often difficult to justify and may not hold in practice.

To address these challenges of residual nonlinearity and encoder–decoder mismatch in a principled, data-driven manner, we propose a framework that integrates conformal prediction into *closed-loop* Koopman-based control. Conformal prediction is a statistical technique that provides distribution-free, finite-sample guarantees on prediction errors under the mild assumption of exchangeable data [9], [10]. We leverage this technique to move beyond empirical, open-loop prediction, formally quantifying the combined effect of these modeling errors on the closed-loop system. This provides a statistically rigorous probabilistic bound on the tracking error induced by a Lyapunov-based feedback control, even with the imperfectly linearized latent space. The connection between latent-space stability and stability of original discrete-time nonlinear systems is also established through the lens of contraction theory. Our main contributions are as follows:

- (1) We propose a statistical method to quantify the closed-loop tracking error in Koopman-based control, capturing residual nonlinearity and encoder–decoder mismatch via distribution-free guarantees of conformal prediction.
- (2) We establish a quantitative guideline for controller design by explicitly relating the closed-loop tracking

performance to key system parameters (e.g., control gains and model uncertainty), enabling the theoretically formal design of learning-based Koopman controllers.

- (3) We apply this method to provide formal, probabilistic guarantees on tracking performance for discrete-time nonlinear systems controlled by Lyapunov-based feedback, connecting latent-space stability with stability in the original systems via contraction theory.
- (4) We validate our framework both in simulations and hardware experiments on a flapping-wing robot (*Flapper Nimble+* from *Flapper Drones*, <https://flapperdrones.com/>), demonstrating its practical applicability in providing reliable performance guarantees for highly nonlinear systems, where our controller serves as a high-level tracking layer that outputs target positions to an onboard PID.

*Related Work:* The Koopman operator provides a linear but infinite-dimensional representation of nonlinear dynamical systems. For their practical applications, approximations in finite-dimensional subspaces are typically employed—e.g., via extended dynamic mode decomposition (eDMD) or neural networks—which enable the use of linear techniques in the analysis and control of nonlinear dynamical systems. For instance, [2] proposed deep autoencoders to learn Koopman-invariant subspaces and apply linear MPC in the latent space. Robustness to modeling errors has been studied in [11], where a robust MPC is constructed under the assumption that the residual modeling error is Lipschitz continuous. However, this assumption can be restrictive, as strong nonlinearities may arise from both the underlying system dynamics and the neural networks often used to learn the lifting functions.

Probabilistic finite-sample error bounds were proposed in [12], offering some theoretical guarantees for control systems approximated via Koopman embeddings with eDMD. Other approaches to compensating for the imperfect linearity include, but are not limited to, tube-based MPC [7], polytopic uncertainty modeling [8], and stochastic methods that model uncertainty explicitly in the latent space [13]. However, many of these approaches rely on structural or distributional assumptions about the model prediction error. This limits their applicability in safety-critical scenarios where such assumptions may not be entirely valid.

Conformal prediction [9], [10] has recently emerged as a powerful framework for uncertainty quantification in data-driven systems. It operates under the assumption of exchangeability, i.e., the joint probability distribution of sampled data is invariant under permutations, thereby relaxing the typical i.i.d. requirement. When applied to dynamical system modeling, this property eliminates the need for structural or distributional priors, making it a promising tool for designing interpretable data-driven control [14]. This concept has been applied in diverse areas of control theory, including safe motion planning [15]–[17], robust control of linear systems [18], [19], safe learning-based MPC [20]–[22], and Lyapunov-based analysis [23]–[25].

Conformal prediction is also explored in Koopman-based

control [26] to derive a statistical bound on one-step ahead state prediction errors in Koopman-based MPC. While it addresses modeling error in the state space in an open-loop, single-step prediction setting, our work focuses on the closed-loop, trajectory-wise control performance degradation caused by imperfectly learned latent spaces. In particular, we explicitly consider aggregated errors from residual nonlinearity and the encoder–decoder mismatch, the latter of which arises from the decoder being an imperfect inverse of the encoder. Our key contribution is to establish a formal statistical characterization of closed-loop tracking error and robustness in such imperfectly lifted spaces. This is supported by its hardware validation with Lyapunov-based control, with the connection between latent-space stability and stability of original systems implied via contraction theory.

## II. PRELIMINARIES AND PROBLEM FORMULATION

We consider the following discrete-time nonlinear control system indexed by the time step  $k$ :

$$x_{k+1} = f(x_k, u_k), \quad (1)$$

where  $x_k \in \mathbb{R}^n$  is the state,  $u_k \in \mathbb{R}^m$  is the control input, and  $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is an *unknown* nonlinear function.

### A. Koopman Operator Framework

The Koopman operator is a linear operator that describes the evolution of observation functions of the state. By lifting the state to a higher-dimensional space of observables, the nonlinear dynamics in the original state space can be represented by linear dynamics in the lifted space.

Let  $\mathcal{G}$  be the space of observable functions  $g : \mathbb{R}^n \rightarrow \mathbb{R}$ . The Koopman operator  $\mathcal{K}$  acts on an observable  $g \in \mathcal{G}$  as:

$$(\mathcal{K}g)(x) = g(f(x, u)).$$

In practice, we use a finite-dimensional approximation of the Koopman operator. We choose a vector of  $N$  basis functions  $\hat{g} : \mathbb{R}^n \rightarrow \mathbb{R}^N$  to lift the state  $x_k$  to a higher-dimensional state  $z_k = \hat{g}(x_k)$ . This allows us to approximate the system (1) by a linear system with a residual error term.

*Definition 1 (Koopman Lifted System):* The system (1) is approximated by the following lifted linear system with a residual error  $d(x_k, u_k)$ :

$$z_{k+1} = Az_k + Bu_k + d(x_k, u_k) \quad (2)$$

where  $z_k = \hat{g}(x_k) \in \mathbb{R}^N$  is the lifted state. The matrices  $A \in \mathbb{R}^{N \times N}$  and  $B \in \mathbb{R}^{N \times m}$  are learned from data such that the pair  $(A, B)$  is controllable.

This approximation enables the full utilization of linear control theory for analyzing and designing control policies. The residual term  $d(x_k, u_k)$  denotes the difference between the lifted next observed state  $\hat{g}(x_{k+1})$  and the linear propagation of the previous observed state  $(Az_k + Bu_k)$ . This will be quantified using conformal prediction in our method.

## B. Split Conformal Prediction

Given a dataset split into a training set and a calibration set, let us consider a point predictor  $\hat{m}(\cdot)$  fitted on the training set. For each point  $(x_i, y_i)$  in the calibration set, we compute the nonconformity score as

$$R_i := \|y_i - \hat{m}(x_i)\|, \quad i = 1, \dots, m.$$

Let  $R_{(1)} \leq \dots \leq R_{(m)}$  be the ordered residuals and define  $k := \lceil (m+1)(1-\alpha) \rceil$ . If the calibration scores are exchangeable with the score of a newly acquired point  $(x_{m+1}, y_{m+1})$ , then we have the following [9], [10]:

$$\Pr[Y_{m+1} \in \hat{C}(X_{m+1})] \geq \frac{\lceil (m+1)(1-\alpha) \rceil}{m+1} \geq 1 - \alpha$$

where the prediction set is given by

$$\hat{C}(x_{n+1}) = [\hat{m}(x_{n+1}) - R_{(k)}, \hat{m}(x_{n+1}) + R_{(k)}].$$

## III. MAIN CONTRIBUTIONS

In this section, we present our primary result: a robust tracking controller and a predictive planner designed in the Koopman lifted space. We first analyze the connection between stability in the Koopman lifted space and stability in our original system using contraction theory. We then delineate the controller design and derive high-probability bounds on the tracking error. Finally, we derive high-probabilistic guarantees on safety.

### A. Controller Design

We design two types of tracking controllers in the lifted system (2). Here, we denote a reference trajectory by  $(x_{d,k}, u_{d,k})$ , with its lifted counterpart in the latent space computed by  $z_{d,k} = \hat{g}(x_{d,k})$ . The subscript  $d$  is used to denote the desired states and inputs. The target system is represented as:

$$z_{d,k+1} = Az_{d,k} + Bu_{d,k} + d'_k(z_{d,k}, u_{d,k}) \quad (3)$$

where  $d'_k : \mathbb{R}^p \times \mathbb{R}^m \mapsto \mathbb{R}^p$  is an unknown function denoting the difference between the lifted next target state  $\hat{g}(x_{d,k+1})$  and the linear propagation of the previous target state.

1) *Nominal Feedback Controller*: Let us first define a Lyapunov-based feedback controller for trajectory tracking.

*Definition 2 (Nominal Feedback Controller (NFC))*: Our nominal feedback controller is defined as:

$$u_k = u_{d,k} - Ke_k. \quad (4)$$

Here, the gain matrix  $K$  is designed such that for a given positive definite matrix  $M = \Theta^T \Theta \succ 0$  satisfying  $\underline{m}\mathbb{I} \preceq M \preceq \bar{m}\mathbb{I}$ , the closed-loop matrix  $A_{cl} = A - BK$  satisfies the following Lyapunov condition:

$$A_{cl}^T M A_{cl} \preceq \gamma M \quad (5)$$

for a given decay rate  $\gamma \in (0, 1)$ .

By defining  $\hat{d}_k(z_k, u_k, z_{d,k}, u_{d,k}) = d_k(z_k, u_k) - d'_k(z_{d,k}, u_{d,k})$ , the dynamics of the tracking error  $e_k := z_k - z_{d,k}$  can be written as:

$$e_{k+1} = A_{cl}e_k + \hat{d}_k(z_k, u_k, z_{d,k}, u_{d,k}). \quad (6)$$

2) *Controller with Robust Disturbance Rejection*: While the nominal controller ensures stability for the nominal system, its performance can be degraded by the residual dynamics  $\hat{d}_k$ . To enhance robustness, we formulate a controller that explicitly mitigates such disturbances by solving an optimization problem at each time step. This is to maintain stability while minimizing deviation from a desired control input. We now define the Controller with Robust Disturbance Rejection (CRDR).

*Definition 3 (CRDR)*: The control input  $u_k$  is determined by solving the following convex optimization problem at each time step  $k$ :

$$(u_k, \Delta_{v,k}) = \arg \min_{(u_k, \Delta_{v,k})} \|\Delta u_k\|^2 + c_v \Delta_{v,k}^2 \quad (7)$$

$$\text{s.t.} \quad \|\Theta(Ae_k + B\Delta u_k)\| \leq \gamma \|\Theta e_k\| - \rho + \Delta_{v,k}$$

where  $\Delta u_k := u_k - u_{d,k}$ . Here,  $v_k := \|\Theta(z_k - z_{d,k})\|$  is a Lyapunov function, with  $\Theta$  obtained by solving the Lyapunov inequality for linear systems in (5). The constraint (7) enforces contraction on the nominal dynamics. The positive constant  $\rho$  serves as a robustness margin against the disturbance  $\hat{d}_k$ , while the slack variable  $\Delta_{v,k}$  ensures feasibility of the optimization even under unforeseen disturbances. The coefficient  $c_v > 0$  is a penalty weight on the slack variable.

### B. Connection to Contraction Theory

Our control design is founded on the principle that stabilizing the system in the lifted space robustly enforces stability in the original nonlinear state space. This is formalized by the equivalence between Koopman stability and contraction, as established in [27] at least for continuous-time systems. We extend their work to discrete-time systems with the nominal feedback controller defined in Definition 4.

*Theorem 1*: If the Jacobian of the lifting map,  $\hat{G}(x_k) = \frac{\partial \hat{g}}{\partial x}(x_k)$ , has full column rank, then the matrix  $W(x_k) = \hat{G}(x_k)^T M \hat{G}(x_k)$  is a valid contraction metric for the original nonlinear system, i.e.,

$$\frac{\partial f_{cl}}{\partial x}(x_k)^T W(f_{cl}(x_k)) \frac{\partial f_{cl}}{\partial x}(x_k) \preceq \gamma W(x), \quad \forall x_k \in \mathbb{R}^n. \quad (8)$$

*Proof*: Let  $f_{cl}(x_k) := f(x_k, u_k) = x_{k+1}$  where  $u_k$  is given by (4). From the definition of linear dynamics in the latent space, the following equation holds:

$$e_{k+1} = A_{cl}e_k, \quad \hat{g}(x_{k+1}) - z_{d,k+1} = A_{cl}(\hat{g}(x_k) - z_{d,k}).$$

Differentiating both sides with respect to  $x$ , we get

$$\hat{G}(f_{cl}(x_k)) \frac{\partial f_{cl}}{\partial x}(x_k) = A_{cl} \hat{G}(x_k)$$

which implies

$$\begin{aligned} & \frac{\partial f_{cl}}{\partial x}(x_k)^T W(f_{cl}(x_k)) \frac{\partial f_{cl}}{\partial x}(x_k) \\ &= \frac{\partial f_{cl}}{\partial x}(x_k)^T \hat{G}(x_k)^T M \hat{G}(x_k) \frac{\partial f_{cl}}{\partial x}(x_k) \\ &= G(x)^T A_{cl}^T M A_{cl} G(x_k). \end{aligned} \quad (9)$$

Since multiplying  $\hat{G}(x)^T$  from the left side and  $\hat{G}(x)$  from the right side on (5) gives  $\hat{G}(x_k)^T A_{cl}^T M A_{cl} \hat{G}(x_k) \preceq \gamma \hat{G}(x_k)^T M \hat{G}(x_k)$ , the relation (9) results in the inequality (8) as desired. ■

### C. High-Probability Error Bounds in Latent Space

While the nominal controller ensures stability for the idealized, error-free system, its performance in practice is heavily affected by the residual dynamics error  $d(x_k, u_k)$ . To guarantee system safety and robustness, we leverage conformal prediction to establish high-probability bounds on this residual error in the latent space. The choice of the nonconformity score, a key element in conformal prediction, depends on the specific controller being analyzed. In the following, we define two distinct nonconformity scores: one for the nominal feedback controller in Definition 2 and another for the robust controller in Definition 3.

*Definition 4 (Forward Nonconformity Score for NFC):*

Let  $\Delta_{d,k} := \|\hat{d}_k(z_k, u_k, z_{d,k}, u_{d,k})\|$  for the residual error in (6). The nonconformity score is defined as:

$$\begin{aligned} s_{\text{fwd},k} &:= \Delta_{d,k} = \|d_k(z_k, u_k) - d'_k(z_{d,k}, u_{d,k})\| \\ &= \|\hat{g}(x_{k+1}) - (A\hat{g}(x_k) + Bu_k) \\ &\quad - \hat{g}(x_{d,k+1}) + (A\hat{g}(x_{d,k}) + Bu_{d,k})\|. \end{aligned}$$

Here, the states  $x_k$  and  $x_{k+1}$  are obtained by sampling state transitions from the real-world system. The desired states  $x_{d,k}$  and  $x_{d,k+1}$  are computed from the reference trajectory and corresponding control inputs generated by the planner.

*Definition 5 (Forward Nonconformity Score for CRDR):*

Let  $\Delta_{d,k} := \|\hat{d}_k(z_k, u_k, z_{d,k}, u_{d,k})\|$  for the residual error in (6). The nonconformity score is defined as:

$$s_{\text{fwd},k} = \sqrt{\bar{m}}\Delta_{d,k} + \Delta_{v,k}$$

where  $\sqrt{\bar{m}}$  denotes the maximum singular value of  $\Theta$  in (7). The term  $\Delta_{v,k}$ , defined also in (7), is determined by applying the controller from Definition 3 to a target trajectory sampled from the same distribution as the deployment environment. For practical application to a physical system, a surrogate model of the dynamics can first be identified from collected data. This model is then used to simulate the tracking of desired trajectories, allowing the data sampling of  $\Delta_{v,k}$ .

*Theorem 2:* Suppose that the forward nonconformity scores in Definitions 4 and 5 are exchangeable with the unseen, future score  $s_{\text{fwd}}$  from the real world, i.e., they satisfy the following inequality via conformal prediction:

$$\Pr[s_{\text{fwd}} \leq q_{1-\delta}] \geq 1 - \delta \quad (10)$$

where  $\delta = \frac{\alpha}{K} \in (0, 1)$  is a user-defined failure probability, and  $q_{1-\delta}$  is the  $(1-\delta)$ th quantile of the empirical distribution of past scores. Then the tracking error in latent space is exponentially bounded with a probability of at least  $1 - \alpha$  for each time step  $k$  as follows:

$$\Pr\left[\|e_{k+K}\| \leq \gamma^K \left(\frac{v_k}{\sqrt{\bar{m}}} - \Delta r\right) + \Delta r\right] \geq 1 - \alpha \quad (11)$$

where for the controller defined in Definition 2,

$$\Delta r := \frac{\sqrt{\bar{m}}q_{1-\delta}}{(1-\gamma)\sqrt{\bar{m}}}$$

for the controller defined in Definition 3,

$$\Delta r := \frac{q_{1-\delta} - \rho}{(1-\gamma)\sqrt{\bar{m}}}.$$

This bound implies that if  $\frac{v_0}{\sqrt{\bar{m}}} - \Delta r \leq 0$ , we have

$$\Pr[\|e_{k+K}\| \leq \Delta r] \geq 1 - \alpha.$$

Here,  $\sqrt{\bar{m}}$  and  $\sqrt{\underline{m}}$  denote the maximum and minimum singular values of  $\Theta$  in (7).

*Proof:* Consider the Lyapunov function in (7). Its time evolution along the dynamics (2) and (3) with the controller of Definition 3 can be computed as

$$\begin{aligned} v_{k+1} &= \|\Theta(z_{k+1} - z_{d,k+1})\| \\ &= \left\| \Theta \left( Ae_k + B\Delta u_k + \hat{d}_k(z_k, u_k, z_{d,k}, u_{d,k}) \right) \right\| \quad (12) \\ &\leq \gamma\|\Theta e_k\| - \rho + \Delta_{v,k} + \|\Theta \hat{d}_k(z_k, u_k, z_{d,k}, u_{d,k})\|. \end{aligned}$$

Since the last term can be bounded as  $\|\Theta \hat{d}_k\| \leq \sqrt{\bar{m}}\|\hat{d}_k\|$ , the inequality (12) can be written as:

$$v_{k+1} \leq \gamma\|\Theta e_k\| - \rho + \Delta_{v,k} + \sqrt{\bar{m}}\hat{d}_k. \quad (13)$$

Unrolling the recursion in (13) from  $k = 0$  to  $K - 1$  yields

$$\begin{aligned} v_K &\leq \gamma^K v_0 + \sum_{k=0}^{K-1} \gamma^{K-1-k} \left( -\rho + \Delta_{v,k} + \sqrt{\bar{m}}\hat{d}_k \right) \\ &\leq \gamma^K v_0 - \rho \frac{1-\gamma^K}{1-\gamma} + \sum_{k=0}^{K-1} \gamma^{K-1-k} s_{\text{fwd},k}. \end{aligned}$$

The probability that (10) holds for all steps  $k = 0, \dots, K - 1$  can be computed by the union bound as follows:

$$\Pr\left[\bigcup_{k=0}^{K-1} s_k > q_{1-\delta}\right] \leq \sum_{k=0}^{K-1} \Pr[s_k > q_{1-\delta}] \leq \sum_{k=0}^{K-1} \frac{\alpha}{K} = \alpha$$

Under this high-probability event, we obtain the following high-probability bound on the Lyapunov function:

$$\Pr\left[v_k \leq \gamma^K v_0 + (q_{1-\delta} - \rho) \frac{1-\gamma^K}{1-\gamma}\right] \geq 1 - \alpha. \quad (14)$$

Finally, we relate the Lyapunov function  $v_k$  back to the tracking error  $\|e_k\|$ . From  $M \succeq \underline{m}\mathbb{I}$ , we get

$$v_K^2 = e_K^T M e_K \geq \underline{m}\|e_K\|^2$$

which implies  $\|e_K\| \leq \frac{v_K}{\sqrt{\underline{m}}}$ . Thus, (14) reduces to

$$\Pr\left[\|e_K\| \leq \gamma^K \frac{v_0}{\sqrt{\underline{m}}} + \frac{q_{1-\delta} - \rho}{(1-\gamma)\sqrt{\underline{m}}}(1 - \gamma^K)\right] \geq 1 - \alpha$$

Using the definition  $\Delta r := \frac{q_{1-\delta} - \rho}{(1-\gamma)\sqrt{\underline{m}}}$ ,

$$\Pr\left[\|e_K\| \leq \gamma^K \left(\frac{v_0}{\sqrt{\underline{m}}} - \Delta r\right) + \Delta r\right] \geq 1 - \alpha$$

By setting  $\rho = \Delta_{v,k} = 0$  in the previous discussion, we can also prove (11) for the nominal feedback controller with  $\Delta r = \frac{\sqrt{\bar{m}}q_{1-\delta}}{(1-\gamma)\sqrt{\bar{m}}}$ . ■

The slack variable  $\Delta_{v,k}$  within the score of Definition 5 is an outcome of the online optimization performed by the controller at each step. Therefore, to obtain a distribution of these scores for conformal prediction before deployment, one must rely on extensive simulations with a surrogate model. We could circumvent this requirement by using the simpler

score from Definition 4 also for the controller in Definition 3. In this case, we can derive an error bound by retaining the explicit summation of the slack variables  $\Delta_{v,k}$ . Specifically, under the high-probability event where  $\|\hat{d}_k\| \leq q_{1-\delta}$  for all  $k \in \{0, \dots, K-1\}$ , we get

$$v_K \leq \gamma^K v_0 + (-\rho + \sqrt{m}q_{1-\delta}) \frac{1-\gamma^K}{1-\gamma} + \sum_{k=0}^{K-1} \gamma^{K-1-k} \Delta_{v,k}.$$

*Corollary 1 (Trajectory-Dependent Error Bound):* When employing CRDR of Definition 3 with the nominal nonconformity score of Definition 4, the probabilistic bound on the tracking error  $\|e_K\|$  is given by:

$$\Pr \left[ \|e_K\| \leq \frac{1}{\sqrt{m}} \left( \gamma^K v_0 + \frac{1-\gamma^K}{1-\gamma} (-\rho + \sqrt{m}q_{1-\delta}) + \sum_{k=0}^{K-1} \gamma^{K-1-k} \Delta_{v,k} \right) \right] \geq 1 - \alpha$$

The error bound is no longer a simple fixed-radius ball, but rather a trajectory-dependent bound that depends on the history of  $\Delta_{v,0}, \dots, \Delta_{v,K-1}$  generated by the controller.

#### D. High-Probability Error Bounds in State Space

Finally, we consider the error bound in the state space when the tracking error in latent space is bounded as in Theorem 2. When the lifting and its inverse mapping are trained separately, e.g., as neural networks, the inverse mapping  $\hat{g}_{\text{inv}}$  may not be equal to the inverse of the lifting function  $\hat{g}^{-1}$ . We define a nonconformity score to quantify such a discrepancy.

*Definition 6 (Round-trip Nonconformity Score):* The ‘round-trip’ nonconformity score is defined as:

$$s_k := \|x_k - \hat{g}_{\text{inv}}(\hat{g}(x_k))\|$$

*Theorem 3 (State-Space Tracking Error Bound):*

Suppose that the following assumptions hold:

- (1) The conditions of Theorem 2 hold, i.e., the tracking error is bounded as  $\Pr[\|e_k\| \leq \epsilon_K] \geq 1 - \alpha$ , where  $\epsilon_K = \gamma^K \left( \frac{v_0}{\sqrt{m}} - \Delta r \right) + \Delta r$
- (2)  $1 - \beta/2$  quantile of the round-trip nonconformity score  $q_{\text{rt}}$  provides the guarantee  $\Pr[\|x_K - \hat{g}_{\text{inv}}(\hat{g}(x_K))\| \leq q_{\text{rt}}] \geq 1 - \beta/2$ .
- (3)  $\hat{g}_{\text{inv}}$  is  $L$ -Lipschitz continuous with constant  $L > 0$ .
- (4) The desired state-space trajectory  $x_{d,K}$  is given, and its lifted-space counterpart is computed as  $z_{d,K} = \hat{g}(x_{d,K})$ .

Then for each time step  $k$ , we have the following:

$$\Pr[\|x_{k+K} - x_{d,k+K}\| \leq 2q_{\text{rt}} + L\epsilon_K] \geq 1 - \alpha - \beta.$$

*Proof:* By applying the triangle inequality, we obtain:

$$\|x_K - x_{d,K}\| \leq \|x_K - \hat{g}_{\text{inv}}(z_K)\| + \|\hat{g}_{\text{inv}}(z_K) - \hat{g}_{\text{inv}}(z_{d,K})\| + \|\hat{g}_{\text{inv}}(z_{d,K}) - x_{d,K}\| \quad (15)$$

The first and third term are the round-trip error for the true state and the desired state. By assumption (2), both terms are independently bounded by the conformal quantile

$q_{\text{rt}}$  each with a probability of at least  $1 - \beta/2$ . Let these be Events A and C. The second term is bounded using the  $L$ -Lipschitz property of  $\hat{g}_{\text{inv}}$  (Assumption (3)). From the premise of Theorem 2, the lifted-space error is bounded by  $\epsilon_K$  with a probability of at least  $1 - \alpha$ . Let this be Event B. The overall bound in inequality (15) holds if Events A, B, and C occur simultaneously. Using the union bound, the probability that at least one of these events fails is at most  $\beta/2 + \alpha + \beta/2 = \alpha + \beta$ . Thus, the joint event occurs with a probability of at least  $1 - \alpha - \beta$ . When this joint event occurs, we substitute the individual bounds into inequality (15):

$$\|x_K - x_{d,K}\| \leq q_{\text{rt}} + L\epsilon_K + q_{\text{rt}} = 2q_{\text{rt}} + L\epsilon_K$$

This inequality holds with a probability of at least  $1 - \alpha - \beta$ , which completes the proof. ■

This analysis provides a crucial insight: the tracking error bound is not solely determined by modeling error. Instead, it is explicitly shaped by key design parameters—the controller’s Lyapunov matrix eigenvalues ( $\bar{m}, \underline{m}$ ) and the decoder’s Lipschitz constant  $L$ . This offers a guideline for designing Koopman controllers, directly linking these fundamental parameters to the achievable tracking precision.

Conditions such as latent contraction, a full-rank Jacobian, and the Lipschitz property of the decoder can be enforced during training through appropriate loss functions. However, in our simulations, these conditions are verified *a posteriori* for simplicity. Our empirical data suggest these conditions are largely satisfied within the calibration region. Enforcing these properties explicitly by design can be an important direction for future work.

## IV. NUMERICAL SIMULATIONS

We first validate our proposed method using the Dubins car model, which exhibits nonholonomic behavior and nonlinear state evolution. The discrete-time dynamics of the Dubins car are given by  $x_{k+1} = x_k + v_k \Delta t \cos \theta_k$ ,  $y_{k+1} = y_k + v_k \Delta t \sin \theta_k$ ,  $\theta_{k+1} = \theta_k + \omega \Delta t$ , and  $v_{k+1} = v_k + a_k \Delta t$ . Here,  $(x, y)$  is the position of the car,  $\theta$  is its heading angle,  $v$  is the forward velocity,  $a$  is the acceleration,  $\omega$  is the angular velocity, and  $(a, \omega)$  is the control input.

### A. Training

The training process is conducted in two sequential phases. The first phase focuses on learning a latent representation, while the second identifies a linear dynamical model within that space.

1) *Data Collection:* Training data is generated in simulation with constant speed and random steering input sampled at 10-step intervals. We collect 1,000 episodes (100 steps each) for representation learning and 100 episodes for dynamics identification.

2) *Phase 1: Latent Representation Learning:* An encoder-decoder architecture is trained to find an effective latent representation of the system dynamics. The encoder,  $g(\cdot)$ , an MLP, maps an observation  $x_k \in \mathbb{R}^4$  (vehicle position and orientation) to a latent state  $z_k \in \mathbb{R}^6$ . A corresponding

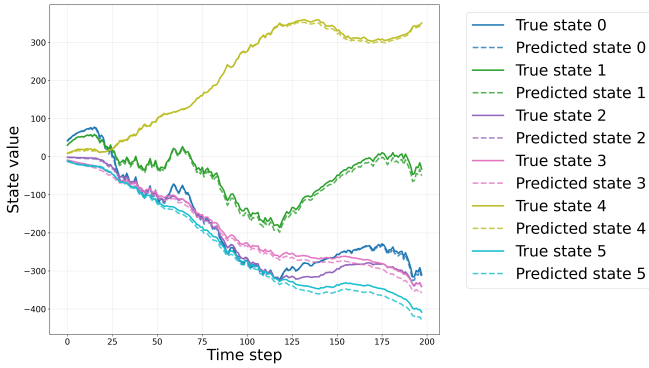


Fig. 2: Latent space one-step prediction. The identified linear model accurately predicts the next latent state from the current state and control input. Here, one-step prediction refers to propagating the latent state by one step and then decoding it back to the original space.

MLP decoder,  $h(\cdot)$ , reconstructs the observation from this latent state. Both networks feature a 256-dimensional hidden layer with ReLU activation and Batch Normalization. This network is trained by minimizing a composite loss function,  $\mathcal{L}_{\text{NN}}$ , designed to encourage accurate reconstruction and dynamically consistent embeddings:

$$\mathcal{L}_{\text{NN}} = \mathcal{L}_{\text{rss}} + \alpha \mathcal{L}_{\text{recon}} + \beta \mathcal{L}_{\text{ctrl}}$$

where  $\mathcal{L}_{\text{rss}}$  is the one-step prediction error in the latent space (using batch-wise linear approximations),  $\mathcal{L}_{\text{recon}} = \mathbb{E}[\|x_k - h(g(x_k))\|_2^2]$  is the autoencoder's reconstruction error, and  $\mathcal{L}_{\text{ctrl}}$  is a controllability penalty.

3) *Phase 2: Linear Dynamics Identification*: Once the encoder-decoder network is trained, the encoder's parameters are frozen. The learned mapping  $z_k = g(x_k)$  is used to transform the entire dataset of observation sequences into a static set of latent state transitions. Subsequently, a single linear dynamical model,  $z_{k+1} = Az_k + Bu_k$ , is identified by finding the matrices A and B that minimize the following loss function,  $\mathcal{L}_{\text{Koopman}}$ :

$$\mathcal{L}_{\text{Koopman}} = \mathcal{L}_{\text{pred}} + w_\rho \mathcal{L}_\rho + w_{\text{ctrl}} \mathcal{L}_{\text{ctrl}}$$

Here,  $\mathcal{L}_{\text{pred}} = \mathbb{E}[\|z_{k+1} - (Az_k + Bu_k)\|_2^2]$  is the primary mean squared prediction error. The loss also includes two regularization terms:  $\mathcal{L}_\rho = \rho(A)$ , which penalizes the spectral radius of A to encourage stability, and the same controllability loss,  $\mathcal{L}_{\text{ctrl}} = -\log(\sigma_{\min}(C) + \epsilon) + \lambda \frac{\sigma_{\max}(C)}{\sigma_{\min}(C) + \epsilon}$ , to ensure that the final linear model is robust and uniformly controllable. The resulting one-step prediction accuracy is shown in Figure 2

The control feedback gain  $K$  for the nominal feedback controller is synthesized by solving a linear matrix inequality (LMI) to theoretically guarantee stability, as confirmed in Figure 3.

### B. Reference Trajectory Construction

We construct a circular reference trajectory  $\{x_{d,k}, u_{d,k}\}_{k=0}^{T-1}$  with radius  $r > 0$ , centered at  $(0, r)$ , starting from the origin with desired speed  $v_d > 0$ . The reference observation  $x_{d,k}$  consists of position  $(x_{\text{ref}}, y_{\text{ref}})$  and heading  $(\sin \theta_d, \cos \theta_d)$ , and the control input  $u_{d,k} = \omega_k$  is the angular velocity computed by finite difference  $\omega_k \approx \frac{\theta_d(t_{k+1}) - \theta_d(t_{k-1})}{2\Delta t}$  and clipped to  $[-\pi, \pi]$ .

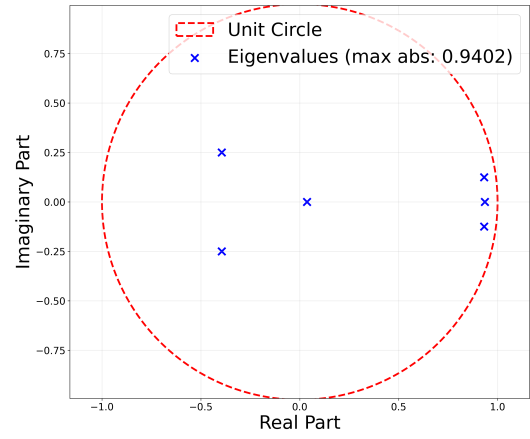


Fig. 3: Closed-loop eigenvalues for the nominal feedback controller. The controller design yields a theoretically stable system, with all eigenvalues located inside the unit circle.

### C. Evaluation

We evaluate the controller performance on the circular trajectory tracking task described previously. The latent representation is trained with loss weights  $\alpha = 0.1$  and  $\beta = 0.1$ . CRDR is designed with parameters  $c_v = 0.01$ ,  $\rho = 0.073$  and  $\gamma = 0.9$ . We compare its performance against the nominal feedback controller (NFC). The tracking evaluation is conducted for a duration of 50 time steps. The controllability loss weight was the most critical hyperparameter: disabling it caused the condition number of the dynamics matrices to exceed  $10^6$ , leading to unstable control. The reconstruction weight was stable across  $[0.05, 0.2]$ , and the embedding dimension  $N \approx 1.3n$  balanced expressiveness and controllability.

### D. Results

The results are presented in Figure 4. Despite its theoretical stability, NFC consistently fails due to control inputs exceeding the actuator saturation limits ( $\pm\pi$  rad/s), causing unbounded error accumulation. In contrast, CRDR achieves excellent tracking by finding inputs that satisfy Lyapunov stability conditions while compensating for the modeling error. Furthermore, the position error of CRDR remains well within the trajectory-dependent theoretical bound derived from Corollary 1, validating our error analysis.

## V. EXPERIMENTS WITH A FLAPPING-WING ROBOT

This section presents the experimental validation of our proposed framework using a flapping-wing robot, hereafter referred to as the Flapper Drone [28] (*Flapper Nimble+* from *Flapper Drones*, <https://flapper-drones.com/>). The experiments are designed to 1) validate the distribution-free error bounds derived using our method, and 2) demonstrate the improved trajectory tracking performance compared to a built-in PID controller.

### A. Experimental Setup

The Flapper Drone is characterized by highly nonlinear aerodynamic and inertial properties, making it a challenging and suitable platform for evaluating data-driven control

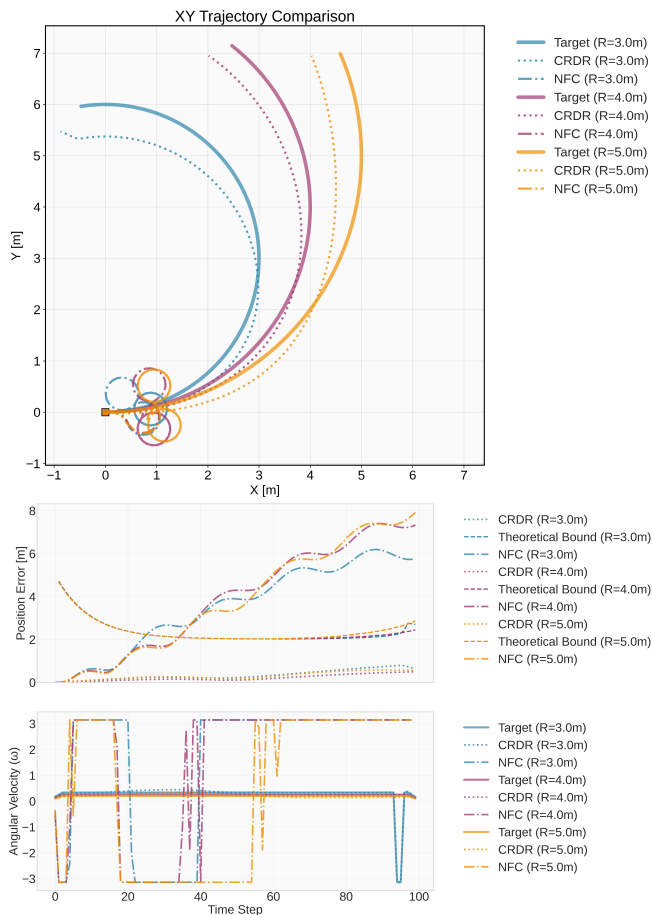


Fig. 4: Controller performance comparison. (Top) XY trajectory tracking. (Bottom) Position error with theoretical bounds and the corresponding angular velocity control input ( $\omega$ ) over time.

methodologies. The drone is equipped with a manufacturer-provided PID controller for low-level attitude and position stabilization. Our approach introduces a high-level, Koopman-based control law that provides target position commands,  $\mathbf{u}_k$ , integrated into the underlying PID. We treat the onboard PID controller as part of the unknown nonlinear dynamics,  $f$ , aiming to model:  $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \text{PID}(\mathbf{u}_k))$ ,  $\mathbf{z}_k = \phi(\mathbf{x}_k)$ ,  $\mathbf{z}_{k+1} = A\mathbf{z}_k + B\mathbf{u}_k$ , and  $\hat{\mathbf{x}}_{k+1} = \psi(\mathbf{z}_{k+1})$ . Here,  $\phi$  and  $\psi$  are the lifting and decoding functions, respectively,  $\mathbf{x}_k$  is the drone's state, and our high-level control input  $\mathbf{u}_k = (x_{d,k}, y_{d,k}, z_{d,k})$  commands the target position. The primary objective is to accurately track circular trajectories with a radius of 0.5 m. This radius ensures the drone remains within the reliable tracking volume of our motion capture system while allowing sufficient margin for the theoretical error bound.

### B. Data Collection and Model Training

To learn the system dynamics, we first collect a dataset of the drone's position and orientation. This is achieved by commanding the drone to track a predefined circular trajectory with a radius of 0.5 m using its default PID controller. From this dataset, we train a neural network as a surrogate model to predict the one-step-ahead state,  $\mathbf{x}_{k+1}$ , given the current state,  $\mathbf{x}_k$ , and a target position,

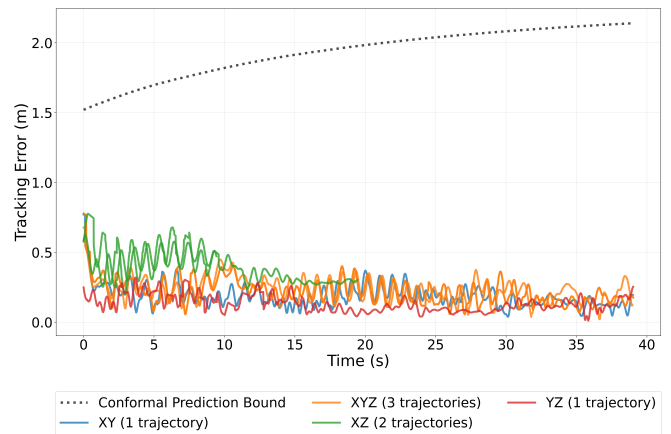


Fig. 5: Tracking error of the Koopman-based controller for various trajectories. The measured error (solid lines) consistently stays below the theoretically computed bound (dotted line), validating our assured error bound estimation. For a visual illustration of the spatial trajectories in these planes, see the YZ-plane example in Fig. 6.

$\mathbf{u}_k$ . Subsequently, the Koopman operator for the system is learned from the data using our proposed methodology. The architecture follows an encoder ( $\phi$ :  $12 \rightarrow 1024 \rightarrow 16$ ), linear dynamics ( $\mathbf{z}_{k+1} = A\mathbf{z}_k + B\mathbf{u}_k$ ,  $A \in \mathbb{R}^{16 \times 16}$ ,  $B \in \mathbb{R}^{16 \times 3}$ ), and decoder ( $\psi$ :  $16 \rightarrow 1024 \rightarrow 12$ ) pipeline with ReLU and batch normalization. The model uses only the current state  $\mathbf{x}_k$  (12-dim: 3D position and flattened rotation matrix) and input  $\mathbf{u}_k$ , without stacking past states, yielding sufficient prediction accuracy.

### C. Validation of the Assured Tracking Bound

To calculate the theoretical tracking error bound, we first generate a calibration dataset for conformal prediction. This is achieved by using the trained surrogate model to simulate CRDR. CRDR is designed with parameters  $c_v = 100$ ,  $\rho = 1.0$  and  $\gamma = 0.9$ . During this simulation, we sample the necessary data to estimate the quantiles for our error bound calculation. From the learned Koopman dynamics, the singular values of  $\Theta$  were calculated as  $\sqrt{\overline{m}/\underline{m}} = 1.00$ , which suggests isotropic convergence in latent space. The calibration process for conformal prediction yields a forward prediction quantile of  $q_{\text{fwd}} = 0.1365$  and a round-trip quantile of  $q_{\text{rt}} = 0.4691$ . These values are then used to compute the time-varying theoretical tracking error bound. Finding stabilizing controller gains required significant tuning effort; we utilized the learned surrogate model to simulate and optimize these parameters offline, avoiding hardware risk.

To validate this bound, we conduct flight experiments where the drone tracks circular trajectories in various planes. Figure 5 shows the real-time tracking error from these experiments. It indicates that the tracking error consistently remains below the computed theoretical bound for all trajectories, thereby implying the validity of our approach.

### D. Improved Tracking Performance

Finally, we compare tracking performance against the default PID controller on a YZ-plane circular trajectory (Figure 6). The PID controller exhibits significant oscillations,

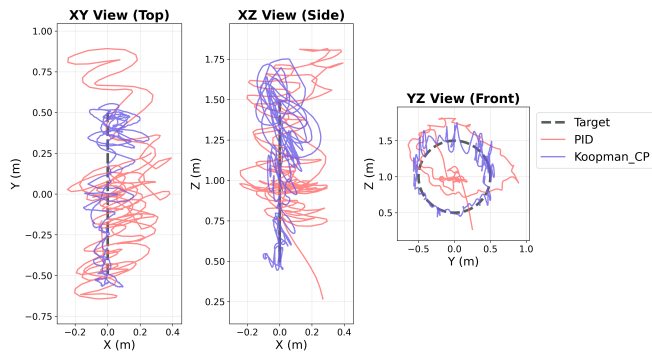


Fig. 6: Comparison of tracking performance for a circular trajectory in the YZ plane. The proposed Koopman-based controller (Koopman\_CP, purple) demonstrates significantly reduced oscillations and superior tracking accuracy compared to the default PID controller (red).

while our Koopman-based controller tracks the reference with markedly higher precision, demonstrating the effectiveness of our data-driven approach.

## VI. CONCLUSION

We present a framework that integrates conformal prediction with *closed-loop* Koopman-based control of discrete-time nonlinear systems, with its theoretical connection to incremental stability in contraction theory. It provides distribution-free tracking error bounds by quantifying latent space prediction errors without assuming a specific error structure. Our analysis shows that the tracking error depends critically on not just latent modeling error, but also the feedback gain and the decoder’s Lipschitz constant, offering a key control design guideline. The framework was validated in simulations with a Dubins car and hardware experiments with a flapping-wing robot, where measured tracking errors consistently stayed within our theoretical bounds.

Future work will aim to reduce the bounds’ conservativeness and address exchangeability challenges when tracking diverse trajectories. In particular, our safety guarantees are valid only within the calibration data distribution; out-of-distribution large disturbances or hardware degradation could invalidate them. Extending the evaluation to diverse trajectory families and benchmarking against other high-level planners is also an important future direction.

## REFERENCES

- [1] M. Korda and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” *Automatica*, vol. 93, pp. 149–160, 2018.
- [2] B. Lusch, J. N. Kutz, and S. L. Brunton, “Deep learning for universal linear embeddings of nonlinear dynamics,” *Nature Communications*, vol. 9, no. 1, pp. 1–10, 2018.
- [3] N. Takeishi, Y. Kawahara, and T. Yairi, “Learning Koopman invariant subspaces for dynamic mode decomposition,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [4] C. Wehmeyer and F. Noé, “Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics,” *The Journal of Chemical Physics*, vol. 148, no. 24, 2018.
- [5] J. Morton, A. Jameson, M. J. Kochenderfer, and F. Witherden, “Deep dynamical modeling and control of unsteady fluid flows,” in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [6] G. Mamakoukas, S. Di Cairano, and A. P. Vinod, “Robust model predictive control with data-driven Koopman operators,” in *American Control Conference*. IEEE, 2022, pp. 1895–1902.

- [7] X. Zhang, W. Pan, R. Scattolini, S. Yu, and X. Xu, “Robust tube-based model predictive control with Koopman operators,” *Automatica*, vol. 137, p. 110114, 2022.
- [8] M. Wang, X. Lou, and B. Cui, “Learning-based robust model predictive control with data-driven Koopman operators,” *International Journal of Machine Learning and Cybernetics*, vol. 14, pp. 3295–3321, 2023.
- [9] V. Vovk, A. Gammerman, and C. Saunders, “Machine-learning applications of algorithmic randomness,” in *International Conference on Machine Learning*, 1999, p. 444–453.
- [10] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer, 2022.
- [11] G. Mamakoukas, S. D. Cairano, and A. P. Vinod, “Robust model predictive control with data-driven Koopman operators,” in *American Control Conference*. IEEE, 2022, pp. 1895–1902.
- [12] F. Nüske, S. Peitz, F. Philipp, M. Schaller, and K. Worthmann, “Finite-data error bounds for Koopman-based prediction and control,” *Journal of Nonlinear Science*, vol. 33, no. 14, 2023.
- [13] M. Han, J. Euler-Rolle, and R. Katzschmann, “Desko: Stability-assured robust control with a deep stochastic Koopman operator,” in *International Conference on Learning Representations*, 2022.
- [14] L. Lindemann, Y. Zhao, X. Yu, G. J. Pappas, and J. V. Deshmukh, “Formal verification and control with conformal prediction,” *arXiv preprint*, vol. arXiv:2409.00536, 2024.
- [15] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, “Safe planning in dynamic environments using conformal prediction,” *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5116–5123, 2023.
- [16] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, “Adaptive conformal prediction for motion planning among dynamic agents,” in *Learning for Dynamics & Control Conference*, 2023, pp. 300–314.
- [17] J. Sun, Y. Jiang, J. Qiu, P. Nobel, M. J. Kochenderfer, and M. Schwager, “Conformal prediction for uncertainty-aware planning with diffusion dynamics model,” in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 80 324–80 337.
- [18] Y. Patel, S. Rayan, and A. Tewari, “Conformal robust control of linear systems,” *arXiv preprint*, 2024, arXiv:2405.16250.
- [19] E. E. Vlahakis, L. Lindemann, P. Sotasakis, and D. V. Dimarogonas, “Conformal prediction for distribution-free optimal control of linear stochastic systems,” *IEEE Control Systems Letters*, vol. 8, p. 2835–2840, 2024.
- [20] K. Y. Chee, M. A. Hsieh, and G. J. Pappas, “Uncertainty quantification for learning-based mpc using weighted conformal prediction,” in *IEEE Conference on Decision and Control*, 2023, pp. 342–349.
- [21] K. Y. Chee, T. C. Silva, M. A. Hsieh, and G. J. Pappas, “Uncertainty quantification and robustification of model-based controllers using conformal prediction,” in *Learning for Dynamics & Control Conference*, vol. 242, July 2024, pp. 528–540.
- [22] H. Zhou, Y. Zhang, and W. Luo, “Safety-critical control with uncertainty quantification using adaptive conformal prediction,” in *American Control Conference*, 2024, pp. 574–580.
- [23] S. Yang, G. J. Pappas, R. Mangharam, and L. Lindemann, “Safe perception-based control under stochastic sensor uncertainty using conformal prediction,” in *IEEE Conference on Decision and Control*, 2023, pp. 6072–6078.
- [24] T.-W. Hsu and H. Tsukamoto, “Statistical guarantees in data-driven nonlinear control: Conformal robustness for stability and safety,” *IEEE Control Systems Letters*, vol. 9, p. 997–1002, 2025.
- [25] S. Wei, M. Ornik, and H. Tsukamoto, “Conformal contraction for robust nonlinear control with distribution-free uncertainty quantification,” in *IEEE Conference on Decision and Control*, to appear, Dec. 2025. [Online]. Available: <https://arxiv.org/pdf/2507.13613>
- [26] K. Liang, G. Yang, M. Cai, and C.-I. Vasile, “Safe navigation in dynamic environments using data-driven Koopman operators and conformal prediction,” *arXiv preprint*, vol. arXiv:2504.00352, 2025.
- [27] B. Yi and I. R. Manchester, “On the equivalence of contraction and Koopman approaches for nonlinear stability and control,” *IEEE Transactions on Automatic Control*, vol. 69, no. 7, pp. 4336–4351, 2024.
- [28] M. Karásek, F. T. Muijres, C. D. Wagter, B. D. W. Remes, and G. C. H. E. de Croon, “A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns,” *Science*, vol. 361, no. 6407, pp. 1089–1094, 2018.