Linear or Bilinear: A Criterion for Koopman Rollouts in Sampling-Based Predictive Control

Kangmin Lee , Sanghyun Kim*

Abstract: Koopman operator models let sampling-based Model Predictive Path Integral (MPPI) control roll out thousands of sampled input sequences through an inexpensive learned lifted model. Existing Koopman-MPPI, however, keeps that lifted model *linear in the input*, a convention inherited from convex Koopman MPC that sampling never required. It fixes one input response for every configuration, and no criterion says when that is enough. This paper supplies the criterion and builds the controller around it. When the *configuration-dependent gain* through which the input reaches the tracked output varies appreciably over the operating region, no constant input matrix can match it under the standard linear readout, no state dictionary repairs the mismatch, and a *bilinear* lift becomes necessary. When the gain is nearly constant, the two lifts predict alike. Since a bilinear step stays in the same matrix–vector cost regime either way, the criterion makes the bilinear model the default rollout. Each sample's cost carries an online error tube from a proportional identification bound, and a free-energy analysis supplies a practical-stability guarantee. Across six systems, bilinear Koopman MPPI tracks with near-oracle accuracy on a wheeled vehicle and a drone under body-frame velocity commands, where its linear counterpart fails structurally. On force- and acceleration-driven platforms the two rollouts perform alike and bilinearity keeps mainly a parameter-efficiency advantage, which explains the successes reported for linear Koopman control. Hyperparameters and videos: `https://rcilab.github.io/koopman-mppi`.

Keywords: Model predictive path integral control, Koopman operator, bilinear systems, sampling-based MPC.

1. INTRODUCTION

For many robotic platforms, an accurate analytical model is not available. Friction, actuation dynamics, and platform-specific parameters are hard to derive by hand, and a calibrated model rarely stays accurate in deployment. Data-driven predictive control instead identifies the rollout model directly from input–state trajectories. The learned model must then predict the controlled variables accurately, yet stay cheap enough for the online optimization loop.

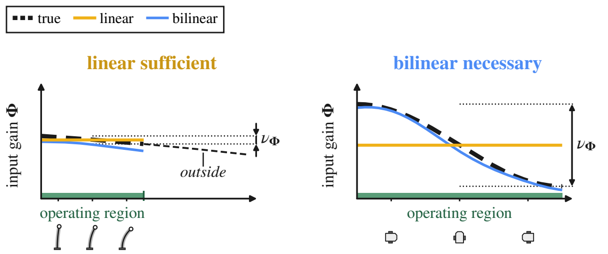
Both demands peak in sampling-based model predictive control (MPC), most prominently Model Predictive Path Integral (MPPI) control [1,2]. MPPI is derivative-free, embarrassingly parallel on GPUs, and free of structural requirements such as convexity, differentiability, or affineness on the rollout dynamics. In exchange, at every control step the controller propagates thousands of sampled input sequences over the horizon, so the rollout model dominates the cost of the control loop [3]. Sampling-based control therefore needs a learned model that pairs control-relevant accuracy with a structured, parallelizable rollout.

Koopman operator models supply exactly this structure. Lifting the state through a dictionary of observables [4–6] renders the dynamics (near-)linear in the lifted space, and the lifted model is identifiable directly from data [7, 8]. Each rollout then reduces to matrix–vector products, far cheaper than integrating the original dynamics. Recent work accordingly reports substantial speedups from MPPI with a learned *linear* Koopman rollout [9].

The *linear* form of that lifted model, however, is a convention imported from a different optimizer. In Koopman MPC the linear predictor keeps the online problem a convex quadratic program, so linearity is forced by convexity, not chosen for the system. Sampling-based control imposes no convexity at all, yet existing Koopman-MPPI methods keep the constant input matrix. Bilinear lifts, in which the input also multiplies the lifted state, are well established in gradient-based Koopman MPC [10, 11], yet neither literature offers a criterion for when the linear choice suffices. *When is a linear Koopman rollout sufficient for sampling-based control, and when is the bilinear form necessary?*

The distinction is a property of the system, not of the optimizer. For a control-affine system $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$, with drift f and input matrix $\boldsymbol{G}(\boldsymbol{x})$, an output map h selects the tracked output $\boldsymbol{y} = h(\boldsymbol{x})$. Over one step Δt under a zero-order hold, the input moves this output through the

Kangmin Lee and Sanghyun Kim are with the Department of Mechanical Engineering, Kyung Hee University, Yongin, South Korea. Sanghyun Kim is also with the Advanced Institute of Convergence Technology, Suwon, South Korea (e-mail: kim87@khu.ac.kr).
* Corresponding author.



(a) (b)

Fig. 1. The gain-variation criterion. (a) A soft arm's slowly varying curvature keeps the input gain nearly constant over the operating region (ν_Φ small). (b) A unicycle's heading sweeps the gain over its full range (ν_Φ large).

first-order response

$$\boldsymbol{y}_{t+1} - \boldsymbol{y}_t = \big(\boldsymbol{d}(\boldsymbol{x}_t) + \boldsymbol{\Phi}(\boldsymbol{x}_t)\,\boldsymbol{u}_t\big)\Delta t + \mathcal{O}(\Delta t^2), \qquad (1)$$

with drift response $\boldsymbol{d}(\boldsymbol{x}) := \nabla h(\boldsymbol{x})\, f(\boldsymbol{x})$ and *input gain* $\boldsymbol{\Phi}(\boldsymbol{x}) := \nabla h(\boldsymbol{x})\, \boldsymbol{G}(\boldsymbol{x})$, where ∇h denotes the Jacobian of h. A linear lifted model with the standard linear readout replaces this true gain with one constant matrix. When $\boldsymbol{\Phi}(\boldsymbol{x})$ varies over the operating region, a single constant matrix cannot reproduce the input response everywhere. The heading rotation of a wheeled vehicle and the body-to-world attitude of a drone are exactly such gains, whereas the slowly varying curvature of a soft arm leaves the gain nearly constant, as shown in Fig. 1. A bilinear lift lets the input matrix vary with the lifted state.

This paper turns the linear-versus-bilinear question into a testable criterion built on the variation of $\boldsymbol{\Phi}(\boldsymbol{x})$ over the operating region. When the variation is large, every linear rollout carries an input-response error that no dictionary choice can remove, and the bilinear form becomes necessary. When it is small, or when the input reaches the output only at second order, the first-order responses coincide and the criterion no longer bars the linear choice. The bilinear model is therefore the natural default, necessary in one regime and harmless in the other. With no convexity constraint to protect, the choice adds essentially no rollout cost, since one bilinear step costs $m+1$ matrix–vector products, where m is the number of inputs, and stays in the rollout regime of the linear model. We then make the default error-aware by propagating the proportional error bound of certified identification [12,13] into an online tube cost with constants from residual fitting.

Our contributions are fourfold:

- **(C1) A testable criterion for the lift.** Under the linear-readout realization, any linear rollout incurs an irreducible first-order input-response error of at least $\frac{1}{2}\nu_\Phi \|\boldsymbol{u}\|\Delta t$ in the worst case, where ν_Φ is the gain variation over the operating region. In closed loop, bilinear Koopman MPPI reaches near-oracle performance where the linear rollout fails systematically.
- **(C2) The failure is structural.** No state dictionary absorbs the coupling $\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{u}$ under a linear readout. Enlarging the linear lift beyond the bilinear model's size does not repair the mismatch.
- **(C3) An error-aware bilinear rollout at matrix–vector cost.** Bilinear Koopman MPPI equips each sample with an online error tube that contains the true lifted error under the certified proportional bound, and it carries a free-energy practical-stability guarantee.
- **(C4) A boundary that organizes prior work.** Force- and acceleration-driven systems with weak gain variation occupy a regime where linear and bilinear predictions largely coincide and bilinearity offers chiefly parameter efficiency. This boundary explains prior linear-Koopman successes [9, 14] under the proposed criterion.

Sec. 2 reviews related work. Sec. 3 develops the criterion, the bilinear rollout, the error tube, and the stability analysis. Sec. 4 validates each claim on six systems, and Sec. 5 concludes with the resulting design rule.

2. RELATED WORK

Rollout models in sampling-based MPC. In its information-theoretic form, MPPI admits arbitrary discrete-time rollout dynamics [1, 2], extending to integrated planning and control stacks for ground vehicles [15]. Much subsequent work improves the *sampler* around a given model, for example through latent-space sampling with an execution-level correction that enforces hard constraints [16]. The rollout model itself spans learned neural networks [1] and full GPU physics engines [3], whose per-sample cost dominates the control step. Model error, where addressed, is absorbed at the sampler level by tube- and robustness-augmented variants [17]. We instead make the model class the design variable, with certified error entering each sample's cost directly.

Linear Koopman predictors and convex MPC. Koopman theory propagates observables of the state linearly [4, 5]. Extended dynamic mode decomposition and its controlled variant identify a finite lifted predictor from data [6, 7], and deep networks learn the dictionary itself [8]. Korda and Mezić turn the linear predictor into an MPC pipeline whose quadratic program is convex by construction [14], and this form has since been the default across Koopman-based control [12], including automated driving [18]. The linear input map is thus an optimizer-driven restriction that buys convexity, not fidelity, and this paper asks what the *system* requires of the lift.

Bilinear models and certified error bounds. With fixed dictionaries, bilinear realizations outperform linear ones in robotic MPC, for example on soft robots [10] and quadrotors [11], at the price of a nonconvex program. In parallel, certified identification supplies proportional error bounds that vanish at the equilibrium and yield closed-loop guarantees in gradient-based MPC [12, 13]. This establishes that bilinearity is *sufficient* and certifiable but leaves open when it is *necessary*, and those guarantees are formulated for gradient-based optimization. We supply both missing pieces: a necessity criterion tied to input-gain variation over the operating region, and incorporation of the proportional bound into the sampling cost with a free-energy practical-stability guarantee.

Koopman rollouts in sampling-based control. The closest work couples MPPI with a learned *linear* deep Koopman rollout and reports substantial speedups on a pendulum, a thrust-driven surface vehicle, and an acceleration-commanded quadruped [9]. There, the linear class is a design convention, and no published criterion identifies when it is adequate. Our criterion places those test systems in the low-variation regime, where the linear choice succeeds, and identifies the velocity-commanded regime, where a constant input matrix structurally fails. We adopt the bilinear default and score it with an error tube built on certified proportional bounds, an error awareness no prior sampling-based Koopman rollout provides.

3. BILINEAR KOOPMAN MPPI

3.1. Problem Statement and Preliminaries

System and objective. We consider a control-affine system $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$, with state $\boldsymbol{x} \in \mathbb{R}^n$, input $\boldsymbol{u} \in \mathbb{R}^m$, drift f, and state-dependent input matrix $\boldsymbol{G}(\boldsymbol{x})$. Discretization with step Δt gives $\boldsymbol{x}_{t+1} = F(\boldsymbol{x}_t, \boldsymbol{u}_t)$, where the transition map F is known only through input–state data $\{(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{x}_{t+1})\}$. Over a horizon T, the controller seeks the input sequence $U = (\boldsymbol{u}_0, \dots, \boldsymbol{u}_{T-1})$ minimizing the cost

$$J(U) = \phi(\boldsymbol{x}_T) + \textstyle\sum_{t=0}^{T-1} q(\boldsymbol{x}_t, \boldsymbol{u}_t), \tag{2}$$

with stage cost q and terminal cost ϕ. Throughout, $\|\cdot\|$ denotes the Euclidean norm and its induced matrix norm.

Information-theoretic MPPI. MPPI draws K input sequences $V^{(k)} = (\boldsymbol{v}_0^{(k)}, \dots, \boldsymbol{v}_{T-1}^{(k)})$, where $\boldsymbol{v}_t^{(k)} = \boldsymbol{u}_t + \varepsilon_t^{(k)}$ and $\varepsilon_t^{(k)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with sampling covariance $\boldsymbol{\Sigma}$. It rolls each sequence through the dynamics, accumulating the cost (2) into a trajectory cost $S(V^{(k)})$, and updates each input using the cost-weighted average of the perturbations [2]:

$$\begin{aligned} \boldsymbol{u}_t &\leftarrow \boldsymbol{u}_t + \frac{\sum_k w^{(k)} \varepsilon_t^{(k)}}{\sum_k w^{(k)}}, \\ w^{(k)} &= \exp\Big(-\tfrac{1}{\lambda}\big(S(V^{(k)}) - \rho\big)\Big), \end{aligned} \tag{3}$$

with temperature $\lambda > 0$ and $\rho = \min_k S(V^{(k)})$. Input penalties, when used, enter through the stage cost q rather than through a separate control-noise term. The only step that scales with the sample count is the per-sample *rollout* evaluating $S(V^{(k)})$.

Koopman lifting and identification. Koopman methods lift the state through a dictionary $\boldsymbol{\Psi} : \mathbb{R}^n \to \mathbb{R}^N$ of lifted dimension N, with $\boldsymbol{\Psi}(\mathbf{0}) = \mathbf{0}$ and a left inverse $\boldsymbol{C}$ ($\boldsymbol{x} = \boldsymbol{C}\,\boldsymbol{\Psi}(\boldsymbol{x})$), and model the evolution of $\boldsymbol{z} = \boldsymbol{\Psi}(\boldsymbol{x})$. The *linear* and *bilinear* predictors are

$$\text{(linear)} \quad \boldsymbol{z}^+ \approx \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{u}, \tag{4}$$

$$\text{(bilinear)} \quad \boldsymbol{z}^+ \approx \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}_0\boldsymbol{u} + \textstyle\sum_{i=1}^m u_i(\boldsymbol{B}_i\boldsymbol{z}), \tag{5}$$

where $\boldsymbol{z}^+$ is the next lifted state, $\boldsymbol{A}$ the state matrix, $\boldsymbol{B}$ and $\boldsymbol{B}_0$ the input matrices, and $\{\boldsymbol{B}_i\}$ the bilinear coupling matrices. Since the lifted input map is state-dependent, a finite *linear* predictor is generically inexact [19], and the *bilinear* form (5) is the minimal extension capturing this dependence when the relevant gains lie in the lifted span. Both predictors are identified by (regularized) least squares via extended dynamic mode decomposition with control [6, 7] for a fixed dictionary, or by joint training of a deep dictionary [8]. The bilinear form (5) simply augments the regressor with the products $u_i\boldsymbol{z}$. Defining $\boldsymbol{M}(\boldsymbol{u}) := \boldsymbol{A} + \sum_i u_i\boldsymbol{B}_i$ gives the bilinear step $\boldsymbol{z}^+ = \boldsymbol{M}(\boldsymbol{u})\boldsymbol{z} + \boldsymbol{B}_0\boldsymbol{u}$, which requires only $m+1$ matrix–vector products and remains in the same rollout regime as the linear model (4).

Proportional error bound. Let $\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{u})$ be the one-step residual of (5). Certified Koopman identification [12, 13] provides, for all $(\boldsymbol{x}, \boldsymbol{u})$ in a certified operating set $\mathcal{X} \times \mathbb{U}$, the *proportional* bound

$$\|\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{u})\| \;\le\; c_x\|\boldsymbol{\Psi}(\boldsymbol{x})\| + c_u\|\boldsymbol{u}\|, \tag{6}$$

whose constants c_x, c_u are estimated from data alongside the model.

3.2. The Criterion: When Bilinearity Is Necessary

Let $\boldsymbol{y} = h(\boldsymbol{x}) \in \mathbb{R}^{n_y}$ be the tracked output of Sec. 1, with first-order response (1) and input gain $\boldsymbol{\Phi}(\boldsymbol{x}) = \nabla h(\boldsymbol{x})\boldsymbol{G}(\boldsymbol{x}) \in \mathbb{R}^{n_y \times m}$. We assume that h is realizable in the lifted space, $h(\boldsymbol{x}) = \boldsymbol{C}_h\boldsymbol{\Psi}(\boldsymbol{x})$, where $\boldsymbol{C}_h \in \mathbb{R}^{n_y \times N}$ has full row rank. For $h(\boldsymbol{x}) = \boldsymbol{\Pi}\boldsymbol{x}$ with a selection matrix $\boldsymbol{\Pi}$, $\boldsymbol{C}_h = \boldsymbol{\Pi}\boldsymbol{C}$. The decisive quantity is the *gain variation* over the operating set $\mathcal{X}$,

$$v_\Phi := \sup_{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}} \big\|\boldsymbol{\Phi}(\boldsymbol{x}) - \boldsymbol{\Phi}(\boldsymbol{x}')\big\|. \tag{7}$$

Proposition 1 (Criterion: a constant input matrix cannot encode a varying gain)**:** Consider one-step output predictions $\boldsymbol{y}_{t+1} = \boldsymbol{C}_h\boldsymbol{z}_{t+1}$ from $\boldsymbol{z}_t = \boldsymbol{\Psi}(\boldsymbol{x}_t)$. (i) For the linear model (4) the input gain $\partial\boldsymbol{y}_{t+1}/\partial\boldsymbol{u}_t = \boldsymbol{C}_h\boldsymbol{B}$ is constant, and for every $\boldsymbol{B}$

$$\sup_{\boldsymbol{x} \in \mathcal{X}} \big\|\Delta t\,\boldsymbol{\Phi}(\boldsymbol{x}) - \boldsymbol{C}_h\boldsymbol{B}\big\| \;\ge\; \tfrac{1}{2}\,v_\Phi\,\Delta t, \tag{8}$$

so the worst-case error of the predicted input response $\boldsymbol{y}_{t+1}(\boldsymbol{u}) - \boldsymbol{y}_{t+1}(\boldsymbol{0})$ over the operating set is at least $\frac{1}{2}\nu_\Phi \|\boldsymbol{u}\|\Delta t - \mathcal{O}(\Delta t^2)$ for a worst-aligned input, irrespective of the dictionary. (ii) For the bilinear model (5) the input gain $\partial \boldsymbol{y}_{t+1}/\partial u_{t,i} = \boldsymbol{C}_h(\boldsymbol{b}_{0,i} + \boldsymbol{B}_i \boldsymbol{z}_t)$ is affine in the lifted state, where $\boldsymbol{b}_{0,i}$ is the ith column of $\boldsymbol{B}_0$. The first-order input response $\Delta t\, \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{u}$ is reproduced exactly whenever every entry of $\boldsymbol{\Phi}$ lies in span$\{1, \Psi_1, \dots, \Psi_N\}$.

Proof: (i) Constancy is immediate from (4). For (8), the triangle inequality gives, for every pair $\boldsymbol{x}, \boldsymbol{x}'$ and every $\boldsymbol{B}$, $\max\left(\|\Delta t\, \boldsymbol{\Phi}(\boldsymbol{x}) - \boldsymbol{C}_h \boldsymbol{B}\|,\ \|\Delta t\, \boldsymbol{\Phi}(\boldsymbol{x}') - \boldsymbol{C}_h \boldsymbol{B}\|\right) \geq \frac{\Delta t}{2}\|\boldsymbol{\Phi}(\boldsymbol{x}) - \boldsymbol{\Phi}(\boldsymbol{x}')\|$. Taking the supremum over pairs yields (8). By (1) the true input response at $\boldsymbol{x}$ is $\Delta t\, \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{u} + \mathcal{O}(\Delta t^2)$ while the model's is $\boldsymbol{C}_h \boldsymbol{B}\boldsymbol{u}$. Choosing $\boldsymbol{u}$ along the top right-singular direction of the mismatch $\Delta t\, \boldsymbol{\Phi}(\boldsymbol{x}) - \boldsymbol{C}_h \boldsymbol{B}$, at a configuration approaching its supremum, gives the stated error. (ii) Differentiating (5) gives the stated gain. If every entry satisfies $\Phi_{ji}(\boldsymbol{x}) = c_{ji} + \boldsymbol{w}_{ji}^\top \boldsymbol{\Psi}(\boldsymbol{x})$, the full row rank of $\boldsymbol{C}_h$ allows $\boldsymbol{B}_0$ and $\{\boldsymbol{B}_i\}$ to be chosen such that $\boldsymbol{C}_h \boldsymbol{b}_{0,i} = \Delta t\,(c_{1i}, \dots, c_{n_y i})^\top$ and $\boldsymbol{C}_h \boldsymbol{B}_i \boldsymbol{\Psi}(\boldsymbol{x}) = \Delta t\,\left(\boldsymbol{w}_{1i}^\top \boldsymbol{\Psi}(\boldsymbol{x}), \dots, \boldsymbol{w}_{n_y i}^\top \boldsymbol{\Psi}(\boldsymbol{x})\right)^\top$. These choices match $\Delta t\, \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{u}$ entrywise. □

Statement (ii) is a representability result. In practice, the identification fits the same matrices jointly to the full lifted evolution.

Remark 1 (No state dictionary can substitute)**:** The coupling $\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{u}$ multiplies the input by a function of the state, while a Koopman dictionary enriches only the *state* representation. Under the standard realization used by linear Koopman-MPPI, a linear readout with a constant input matrix, the input still multiplies the fixed $\boldsymbol{B}$. The lower bound (8) is therefore invariant to the dictionary, deep or fixed. Bilinearity is therefore structural, not a matter of capacity, and the term $u_i(\boldsymbol{B}_i \boldsymbol{z})$ removes exactly this obstruction.

Remark 2 (Reading the criterion)**:** ν_Φ can be computed in advance from known kinematics or estimated from data by regressing local input gains.

Velocity-commanded bases: a body-frame command reaches a world-frame output through a rotation that sweeps its full range as the robot reorients. For the unicycle position the gain is $\boldsymbol{\Phi}(\boldsymbol{x}) \propto (\cos\theta, \sin\theta)^\top$, and for the drone position it is the body-to-world rotation. Hence ν_Φ is maximal, and a constant input matrix is correct at only one configuration.

Force- and acceleration-driven outputs: when the tracked output has relative degree two or higher, $\boldsymbol{\Phi} \equiv \boldsymbol{0}$, $\nu_\Phi = 0$, and the input enters only at $\mathcal{O}(\Delta t^2)$, so a linear lift is not structurally barred.

Because the criterion is defined over the operating set, its relevance depends on the states a task actually visits. The input's operating point enters the same way. Under the split $\boldsymbol{u} = \bar{\boldsymbol{u}} + \delta\boldsymbol{u}$, the component $\boldsymbol{\Phi}(\boldsymbol{x})\bar{\boldsymbol{u}}$ is a pure state function, absorbed by the autonomous dynamics once the entries of $\boldsymbol{\Phi}$ lie in the lifted span, so only $\boldsymbol{\Phi}(\boldsymbol{x})\,\delta\boldsymbol{u}$ is an irreducible state–input product. Hence the thrust-driven quadrotor remains benign even at the velocity level, where its gain does vary. Thrust enters as $T = mg + \delta T$, so the dominant coupling $g\,\boldsymbol{R}\boldsymbol{e}_3$ is state-linear and only the small fluctuation δT passes through the varying gain. A velocity command, by contrast, is zero-centered by construction, so its entire coupling passes through the varying gain. Because the operating-point bias is set by the actuation level, the regime can often be read off before training.

Algorithm 1 Bilinear Koopman MPPI (one control step)

Require: state $\boldsymbol{x}$; warm start $U = (\boldsymbol{u}_0, \dots, \boldsymbol{u}_{T-1})$; model $\boldsymbol{A}, \boldsymbol{B}_0, \{\boldsymbol{B}_i\}, \boldsymbol{C}, \boldsymbol{\Psi}$; norms $\|\boldsymbol{A}\|, \{\|\boldsymbol{B}_i\|\}$; constants c_x, c_u; parameters $K, T, \boldsymbol{\Sigma}, \lambda, \beta, \beta_T$

```
z_0 ← Ψ(x)
for k = 1,...,K in parallel do
    v_t^(k) ← u_t + ε_t^(k), clamped to 𝕌,   ε_t^(k) ~ 𝒩(0, Σ), for all t
    ẑ ← z_0;   e ← 0;   S^(k) ← 0
    for t = 0,...,T−1 do
        S^(k) ← S^(k) + q(C ẑ, v_t^(k)) + β e
        e ← (m̄(v_t^(k)) + c_x) e + c_x ||ẑ|| + c_u ||v_t^(k)||
        ẑ ← M(v_t^(k)) ẑ + B_0 v_t^(k)
    end for
    S^(k) ← S^(k) + φ(C ẑ) + β_T e
end for
w^(k) ← exp(−(1/λ)(S^(k) − min_j S^(j)))
u_t ← u_t + (Σ_k w^(k) ε_t^(k)) / Σ_j w^(j) for all t
return u_0; shift U one step as the next warm start
```

3.3. Bilinear Rollouts at Matrix–Vector Cost

The controller is standard MPPI (3) with every rollout performed by the bilinear surrogate (5). At each control step we lift the state once, $\boldsymbol{z}_0 = \boldsymbol{\Psi}(\boldsymbol{x})$, propagate the K perturbed input sequences through

$$\hat{\boldsymbol{z}}_{t+1} = \boldsymbol{M}(\boldsymbol{v}_t)\,\hat{\boldsymbol{z}}_t + \boldsymbol{B}_0 \boldsymbol{v}_t, \qquad \hat{\boldsymbol{z}}_0 = \boldsymbol{z}_0, \tag{9}$$

score each sample with the task cost plus the tube penalty of Sec. 3.4, and apply the softmax-weighted update (3), as summarized in Alg. 1. One step of (9) costs $m+1$ matrix–vector products of size N against one for the linear model (4).

3.4. An Error Tube in the Sample Cost

Along a sampled sequence $V = (\boldsymbol{v}_0, \dots, \boldsymbol{v}_{T-1})$ the true lifted state evolves as $\boldsymbol{z}_{t+1} = \boldsymbol{M}(\boldsymbol{v}_t)\boldsymbol{z}_t + \boldsymbol{B}_0 \boldsymbol{v}_t + \boldsymbol{r}_t$ with $\boldsymbol{r}_t = \boldsymbol{r}(\boldsymbol{x}_t, \boldsymbol{v}_t)$ bounded by (6). Subtracting (9) shows that the deviation $\boldsymbol{\delta}_t := \boldsymbol{z}_t - \hat{\boldsymbol{z}}_t$ obeys $\boldsymbol{\delta}_{t+1} = \boldsymbol{M}(\boldsymbol{v}_t)\boldsymbol{\delta}_t + \boldsymbol{r}_t$. With the precomputable norm bound $\bar{m}(\boldsymbol{v}) := \|\boldsymbol{A}\| +$

$\sum_{i=1}^{m} |v_i| \, \|\boldsymbol{B}_i\| \;\geq\; \|\boldsymbol{M}(\boldsymbol{v})\|$, define, from $e_0 = 0$,

$$e_{t+1} = \big(\bar{m}(\boldsymbol{v}_t) + c_x\big)\, e_t + c_x \|\hat{\boldsymbol{z}}_t\| + c_u \|\boldsymbol{v}_t\|. \tag{10}$$

The recursion adds one vector norm and a few scalar operations per step (Alg. 1, line 7).

Proposition 2 (Tube bound)**:** Let the true trajectory under V remain in the certified set of (6). Then the tube (10) satisfies $\|\boldsymbol{z}_t - \hat{\boldsymbol{z}}_t\| \leq e_t$ for all t, and hence $\|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\| \leq \|\boldsymbol{C}\|\, e_t$ with $\hat{\boldsymbol{x}}_t := \boldsymbol{C}\hat{\boldsymbol{z}}_t$.

Proof: Induction over t, from the base case $e_0 = 0 = \|\boldsymbol{\delta}_0\|$. Given $\|\boldsymbol{\delta}_t\| \leq e_t$, the deviation dynamics and (6) give $\|\boldsymbol{\delta}_{t+1}\| \leq \|\boldsymbol{M}(\boldsymbol{v}_t)\|\, e_t + c_x\|\boldsymbol{z}_t\| + c_u\|\boldsymbol{v}_t\|$. Substituting $\|\boldsymbol{z}_t\| \leq \|\hat{\boldsymbol{z}}_t\| + e_t$ and $\|\boldsymbol{M}(\boldsymbol{v}_t)\| \leq \bar{m}(\boldsymbol{v}_t)$, with all coefficients nonnegative, yields exactly (10). The state bound follows from $\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t = \boldsymbol{C}\boldsymbol{\delta}_t$. □

The containment inherits the certificate behind (6). Kernel-based certificates are deterministic, and probabilistic certificates at confidence $1 - \delta_c$ keep confidence at least $1 - T\delta_c$ over the horizon by a union bound.

The sampler compares costs, not state errors, so the tube must reach the sample cost. Let $\hat{S}(V;\boldsymbol{x}) = \phi(\boldsymbol{C}\hat{\boldsymbol{z}}_T) + \sum_{t=0}^{T-1} q(\boldsymbol{C}\hat{\boldsymbol{z}}_t, \boldsymbol{v}_t)$ denote the surrogate sample cost and $S(V;\boldsymbol{x})$ its true-trajectory counterpart. If $q(\cdot,\boldsymbol{u})$ and ϕ are Lipschitz with constants L_q (uniformly in $\boldsymbol{u} \in \mathbb{U}$) and L_ϕ, Prop. 2 bounds every sample's cost gap,

$$\big|S(V;\boldsymbol{x}) - \hat{S}(V;\boldsymbol{x})\big| \;\leq\; \Delta_S(V;\boldsymbol{x}), \tag{11}$$

with $\Delta_S(V;\boldsymbol{x}) := \|\boldsymbol{C}\|\big(L_\phi\, e_T + L_q \sum_{t=0}^{T-1} e_t\big)$. Quadratic costs satisfy this requirement locally on the certified set, with L_q proportional to its diameter. We score each sample with the *tube-penalized* cost

$$\tilde{S} := \hat{S} + \beta \sum_{t=0}^{T-1} e_t + \beta_T\, e_T, \tag{12}$$

which the softmax update (3) optimizes in place of $\hat{S}$. Because (6) vanishes at the equilibrium, the penalty steers the search toward the certified region.

Corollary 1 (MPPI on a guaranteed upper bound)**:** With the derived weights $\beta = L_q\|\boldsymbol{C}\|$ and $\beta_T = L_\phi\|\boldsymbol{C}\|$, (12) equals $\hat{S} + \Delta_S \;\geq\; S$. Every sample whose true trajectory remains in the certified set (Prop. 2) is then scored by a guaranteed upper bound on its *true* cost, so the update (3) descends a robust surrogate of the true objective. Smaller β, β_T interpolate toward the risk-neutral controller, and $\beta = \beta_T = 0$ recovers plain bilinear Koopman MPPI.

3.5. Free-Energy Practical Stability

For the sampling distribution π_U (the Gaussian $\mathcal{N}(U,\boldsymbol{\Sigma})$ truncated to $\mathbb{U}^T$, cf. Prop. 3(i)), define the free energy of the true cost $\mathcal{F}(\boldsymbol{x}) = -\lambda \log \mathbb{E}_{V\sim\pi_U}\big[\exp\big(-S(V;\boldsymbol{x})/\lambda\big)\big]$, and $\hat{\mathcal{F}}(\boldsymbol{x})$ analogously using $\hat{S}$. The MPPI update (3) with $\beta = \beta_T = 0$ descends $\hat{\mathcal{F}}$, whereas $\mathcal{F}$ governs the true closed loop.

Proposition 3 (Free-energy practical stability)**:** Assume the following. (i) The certified bound (6) holds on the operating set $\mathcal{X} \times \mathbb{U}$. The sampled sequences are restricted to admissible inputs (sampled from a truncated or clamped Gaussian on the compact set $\mathbb{U}$), and the true rollouts they induce remain in $\mathcal{X}$, an invariance that is assumed rather than established. (ii) The surrogate is cost-controllable on that set, following the surrogate-MPC framework of [12], with stage cost bounded below by $\alpha\|\boldsymbol{\Psi}(\boldsymbol{x})\|^2$, $\alpha > 0$, and admissible inputs bounded proportionally to $\|\boldsymbol{\Psi}(\boldsymbol{x})\|$. (iii) $q(\cdot,\boldsymbol{u})$ is Lipschitz uniformly in $\boldsymbol{u} \in \mathbb{U}$, and ϕ is Lipschitz. (iv) The sampled update attains the surrogate free-energy decrease up to $\delta_\lambda + \delta_K$ with $\delta_\lambda = \mathcal{O}(\lambda)$ and $\delta_K = \mathcal{O}(K^{-1/2})$, the bias of the exponential weighting and the Monte Carlo error of K samples. Then the model-induced free-energy gap obeys

$$\big|\mathcal{F}(\boldsymbol{x}) - \hat{\mathcal{F}}(\boldsymbol{x})\big| \;\leq\; \bar{\Delta}_S(\boldsymbol{x}) := \sup_{V\in\mathbb{U}^T} \Delta_S(V;\boldsymbol{x}), \tag{13}$$

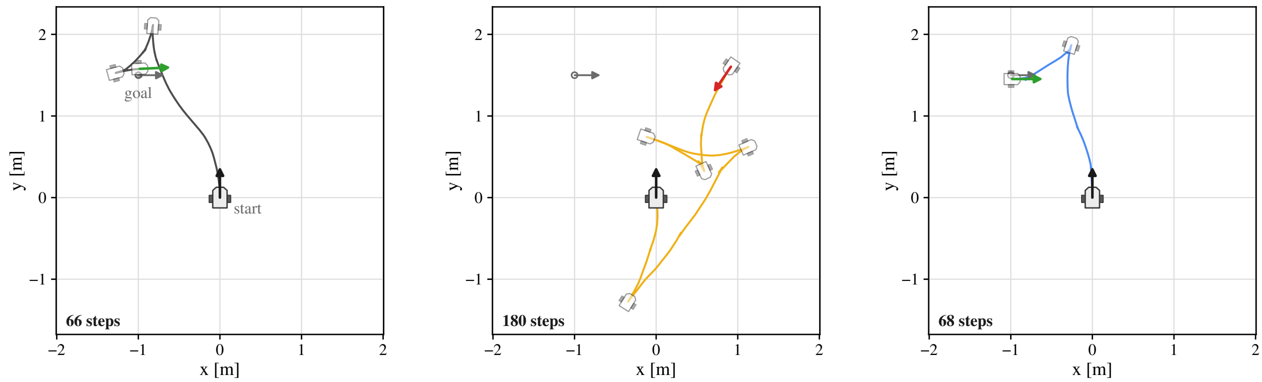
and the true closed loop is practically asymptotically stable with an ultimate bound that vanishes as the model errors (c_x, c_u) and the sampling errors (δ_λ, δ_K) vanish.

Proof sketch: Pointwise, $S \;\geq\; \hat{S} - \Delta_S$ implies $\exp(-S/\lambda) \leq \exp(\bar{\Delta}_S/\lambda)\exp(-\hat{S}/\lambda)$. Monotonicity of the expectation and of $-\lambda\log(\cdot)$ gives $\mathcal{F} \geq \hat{\mathcal{F}} - \bar{\Delta}_S$, and the symmetric inequality gives (13). Assumption (iii) enters only through the cost Lipschitz bound (11), and no smoothness of the value function is required. Cost-controllability (ii) provides a relaxed-dynamic-programming decrease of $\hat{\mathcal{F}}$ along the surrogate [12], degraded by at most $\delta_\lambda + \delta_K$ by (iv). Transporting this decrease to $\mathcal{F}$ through (13) leaves a perturbation of size $\mathcal{O}\big((c_x + c_u)\|\boldsymbol{\Psi}(\boldsymbol{x})\|\big)$ plus the sampling gap, since, along the admissible sequences of (ii), the tube sources $c_x\|\hat{\boldsymbol{z}}_t\| + c_u\|\boldsymbol{v}_t\|$ vanish at the equilibrium by (6), with nonzero input biases recentered by the operating-point split of Remark 2. Outside a sublevel set whose radius vanishes with the model and sampling errors, the quadratic decrease in (ii) dominates this perturbation, yielding practical asymptotic stability with the stated ultimate bound. □

3.6. Identification

Models use the predictor classes and dictionaries of Sec. 3.1, with a learned deep dictionary or a fixed one. The observables are the leading lifted coordinates and $\boldsymbol{C}$ the fixed selector recovering them, so $\boldsymbol{x} = \boldsymbol{C}\boldsymbol{\Psi}(\boldsymbol{x})$ holds by construction. In place of a one-step least-squares fit, we minimize a *multi-step* loss over length-H windows,

$$\mathcal{L} = \sum_{t=1}^{H} \Big(\big\|\boldsymbol{C}\hat{\boldsymbol{z}}_t - \boldsymbol{x}_t\big\|^2 + \gamma\big\|\hat{\boldsymbol{z}}_t - \boldsymbol{\Psi}(\boldsymbol{x}_t)\big\|^2\Big), \tag{14}$$



(a) (b) (c)

Fig. 2. Nonholonomic unicycle parking under (a) the oracle, (b) linear, and (c) bilinear Koopman MPPI.

where $\hat{z}_t$ is rolled from $\hat{z}_0 = \Psi(\boldsymbol{x}_0)$ through (9) and the second term enforces latent consistency. For $m \geq 3$ we initialize $\{\boldsymbol{B}_i\}$ to zero, so training starts from the identified linear model. For $m \leq 2$ a small random initialization is equally stable. For sustained-maneuver deployments, training inputs are drawn as coherent control sequences matching the deployment input distribution.

4. EXPERIMENTS

Setup and mapping to claims. Three controllers are compared: the *linear* baseline [9] and the *bilinear* controller share one deep Koopman lifting (identical network width, training data, and multi-step loss (14)) and differ only in the lifted model, (4) versus (5), while *Oracle* MPPI rolls out the true dynamics. The oracle is not a performance ceiling, since sampling-based MPPI is suboptimal even with the true model. Both learned controllers share the settings of Table 1 and adopt the linear baseline's reported method-specific choices. Unless stated otherwise, the closed-loop studies of Secs. 4.1–4.4 run with $\beta = \beta_T = 0$, and the tube of Sec. 3.4 is evaluated in Secs. 4.5 and 4.6. Drone results are means $\pm$ standard deviation over 60 episodes, and quadrotor results medians over 20 runs. The unicycle is evaluated over 50 parking trials across ten start–goal pairs. The unicycle and drone use standard kinematic models, and the pendulum an analytical model. The quadrotor and torque-controlled arm are simulated in MuJoCo [20], and the soft continuum arm in SoftManiSim [21].

Throughout, the criterion is quantified by the normalized gain variation $\hat{\nu} = \nu_\Phi / \sup_{\boldsymbol{x}} \|\boldsymbol{\Phi}(\boldsymbol{x})\|$, set to zero when $\boldsymbol{\Phi} \equiv \boldsymbol{0}$. Secs. 4.1–4.6 validate, in order, the criterion at maximal gain variation (C1), the capacity question (C2), the negative predictions (C4), and the error tube, rollout cost, and design choices (C3).

Table 1. Key per-system MPPI hyperparameters.

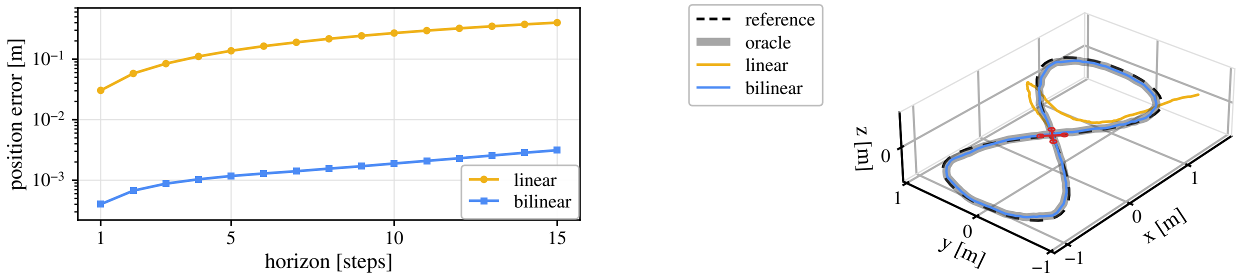
System	N	K	T	Δt (s)	λ
Unicycle	8	1,536	40	0.05	0.4
Drone	13	1,024	15	0.05	0.25
Pendulum	8	2,560	65	0.05	0.8
Quadrotor	24	1,500	20	0.02	0.5

4.1. Where the Criterion Binds

We first test Prop. 1 on two systems whose input gains sweep their full ranges.

Nonholonomic vehicle. A unicycle must park at a goal pose. Its forward-speed input affects position through the heading rotation $(\cos\theta, \sin\theta)$, the maximal-variation case of Remark 2. In open loop the bilinear model predicts position roughly 45 times more accurately than the linear at the planning horizon, and the gap is structural rather than dictionary-driven, since a fixed trigonometric lift reproduces it at roughly 40 times. In closed loop the gap becomes qualitative, as shown in Fig. 2. The bilinear controller parks in 84% of trials, near the oracle (92%), whereas the linear controller parks in 0% of 50, driving off along its initial heading to end 3.1 ± 1.4 m from the goal. As predicted, a constant $\boldsymbol{C}_h\boldsymbol{B}$ is correct at only one heading, so MPPI optimizes an incorrect cost landscape rather than a noisy version of the correct one.

Aerial drone. The drone tests the same mechanism in 3D. Commanded by body-frame velocities, it tracks a figure-8 while yawing along the path, so $\dot{\boldsymbol{p}} = \boldsymbol{R}(\psi)\boldsymbol{v}^{\mathrm{b}}$ and the gain $\boldsymbol{R}(\psi)$ sweeps its full range. In open loop, the bilinear model predicts world position over a hundred times more accurately at the planning horizon, as shown in Fig. 3(a) and Table 2. In closed loop, as shown in Fig. 3(b), the bilinear controller tracks at oracle accuracy across three references and two observation-noise levels (0.046 ± 0.009 m mean error for both, n=60), whereas the



(a) (b)

Fig. 3. 3D velocity-commanded drone. (a) Open-loop position-prediction error over the planning horizon. (b) Closed-loop figure-8 tracking under nominal observation noise.

Table 2. Drone rollout baselines. Bold: best value.

Model	N	Pred. err. (m)
Linear	13	0.409 ± 0.001
Linear (large)	29	0.409 ± 0.001
MLP dynamics	—	0.078 ± 0.017
Bilinear	13	$\mathbf{0.0030 \pm 0.0001}$

linear controller drifts from every path (1.4 ± 1.0 m). By Remark 1 the gap is structural, not one of capacity.

4.2. Ruling Out Capacity

Table 2 tests two objections, that the linear model is under-parametrized and that a generic neural dynamics model would suffice. All rows report the open-loop 15-step world-position error, mean ± standard deviation over ten training runs on a shared held-out set of 1,000 trajectories. More than doubling the linear lift's latent dimension from $N{=}13$ to $N{=}29$ gives it more parameters and rollout FLOPs than the bilinear model, yet leaves its 15-step error unchanged at 0.409 ± 0.001 m over ten training runs. As Remark 1 states, dictionary enrichment leaves the constant input matrix untouched, so no dictionary size can reproduce $\boldsymbol{R}(\psi)$. An unstructured multilayer perceptron (MLP), which can represent the rotation, improves on the linear lift roughly fivefold, yet at a matched training budget it remains more than twenty times less accurate than the bilinear model. The drone is control-affine, so the bilinear lift represents its input response exactly, whereas the MLP must approximate the rotation and forgoes the structure underlying the error tube in Prop. 2.

4.3. Efficiency Where the Coupling Is Weak

In torque-limited pendulum swing-up, the input acts at second order with weak gain variation, so the criterion predicts a linear model is not structurally barred. With a 64-neuron dictionary ($N{=}8$ fixed) the bilinear controller swings up and balances in all five trials, as does the oracle, and it still swings up in four of five trials with a dictionary

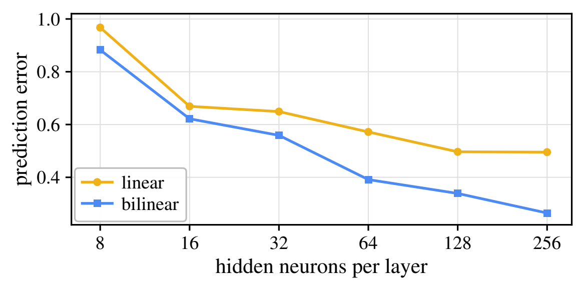


Fig. 4. Pendulum open-loop prediction error against dictionary width ($H{=}12$, $N{=}8$).

as small as 16 neurons. Under the same long-horizon setting, the linear lift does not reliably reach the multi-step accuracy that swing-up demands at the widths we test. The width sweep of Fig. 4 locates the difference in capacity rather than structure. Unlike the structural plateau of Sec. 4.2, the linear model's error falls as the dictionary widens, in line with the width sensitivity reported for the linear baseline [9], yet it saturates above the bilinear asymptote. The bilinear model reaches a given multi-step accuracy with about four times fewer neurons. Even when bilinearity is not structurally required, the structured input term substitutes for network width.

4.4. The Boundary

When the input gain barely varies, the criterion predicts that linear and bilinear rollouts largely coincide.

The thrust-driven quadrotor, shown in Fig. 5, is the clean second-order case. Thrust reaches the position only through acceleration, and although the attitude sweeps aggressive tilts of up to $\pm 69°$, the linear and bilinear predictions coincide over the planning horizon, on the full state and on the control-relevant velocity alike, as the thrust bias of Remark 2 prescribes. At the secondary velocity level, the normalized gain variation is nearly maximal over the full tilt range but drops below 0.25 over the closed-loop states (median tilt $2.7°$). As the crite-

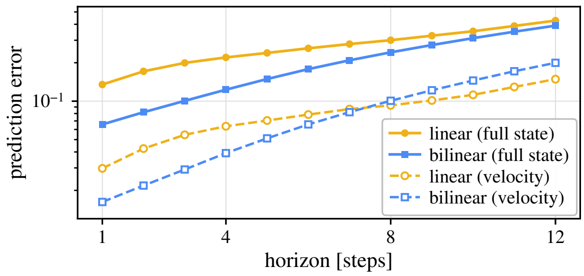


Fig. 5. Thrust-driven quadrotor under aggressive flight.

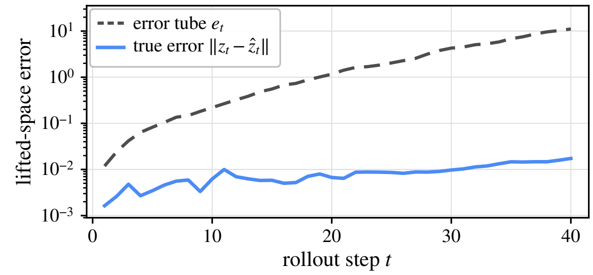


Fig. 6. Error tube e_t (Prop. 2) and the true lifted error on held-out unicycle rollouts.

Table 3. Summary of all systems by normalized gain variation $\hat{\nu}$ (7).

System	Gain variation $\hat{\nu}$	Closed loop: Oracle / Linear / Bilinear	Prediction: bilinear vs. linear
Unicycle	$\boldsymbol{R}(\theta)$, maximal ($\hat{\nu}$=2.0)	92 / 0 / 84 %	42–48×
Drone	$\boldsymbol{R}(\psi)$, maximal ($\hat{\nu}$=2.0)	0.05 / 1.4 / 0.05 m	135 ± 7×
Pendulum	second order ($\hat{\nu}$=0)	—	4× fewer neurons
Quadrotor	second order ($\hat{\nu}$=0)	0.12 / 0.44 / 0.45 m	1.0–1.1×
Torque arm	second order, inertia ($\hat{\nu}$=0)	—	0.88–0.98×
Soft arm	low (curvature)	—	0.93–1.09×

rion predicts, the median landing distance over 20 runs across four goals is 0.45 m (bilinear) and 0.44 m (linear), against 0.115 ± 0.003 m for the true-dynamics oracle. Identification error common to both, not the model class, separates them from the oracle. On a torque-controlled three-link arm (MuJoCo), whose input enters through a configuration-dependent inertia, the two models are indistinguishable in multi-step end-effector prediction across horizons (Table 3), and its data-driven gain estimate stays below the permutation noise floor, consistent with $\hat{\nu} = 0$. A cable-driven soft continuum arm (SoftManiSim, two or four segments, up to twelve cable inputs) shows the same tip-prediction equivalence, in line with reported EDMD-based Koopman success for soft robots [22]. Its slowly varying curvature keeps the effective gain nearly constant, consistent with a deep lift closing the gap that fixed dictionaries leave [10]. This is the regime of the linear baseline's systems, a thrust-driven surface vehicle and an acceleration-commanded quadruped with modest heading changes.

4.5. Error Tube and Rollout Cost

We instantiate the constants c_x, c_u by least squares on held-out residuals of the deep bilinear model, an empirical proxy for certified constants. Under this proxy the tube is sound. Along 2,000 held-out rollouts of the deep bilinear unicycle model (40-step horizons) it contains the true lifted error at every step, with zero violations at all 82,000 evaluated points, as shown in Fig. 6. Tightness is set by the identification, not the tube mechanism. The multi-step-trained model has $\|\boldsymbol{A}\|_2 \approx 1.02$, the residual fit gives $c_x \approx 3.5 \times 10^{-3}$, and certified SafEDMD-grade constants would tighten the same recursion directly.

The bilinear default adds a known constant factor to a linear lifted rollout. Each step requires $m+1$ matrix–vector products rather than one (Sec. 3.3). Measured CPU-only (PyTorch, 4 threads, Intel Core i5-13400F), one bilinear rollout of K=1,024 samples over T=30 (a standalone benchmark) costs 3.9 ms per control step, against 13 ms for a hand-batched true-dynamics integration and 76 ms for a physics-engine rollout.

4.6. Ablations and Design Choices

Zero-initialized bilinear terms. Initializing $\{\boldsymbol{B}_i\}$ at zero starts from the linear model and improves monotonically on it. On input-rich systems ($m \geq 3$) random initialization yields unstable rollouts, whereas either choice is stable for $m \leq 2$.

Multi-step training. One-step regression of either model gives poor long-horizon rollouts, while the H-step loss (14), applied identically to both, makes either usable inside MPPI, so the gaps above are not artifacts of the training objective.

Error-tube weight β. Setting β=0 recovers plain bilinear MPPI, is best for nominal regulation, and is how all closed-loop benchmarks run. With the fitted (uncertified) constants the tube compounds quickly, so any β that influences the sampler lets the tube term dominate the task cost, and a β-sweep on the quadrotor degrades both learned controllers monotonically. The tube therefore serves here as the certificate of Prop. 2, not a practical sample-shaping term.

Table 3 summarizes all six systems. Closed-loop entries report parking rate (unicycle), mean tracking error (drone), and median goal distance (quadrotor). A prediction ratio near $1\times$ means the two rollouts coincide, and dashed rows are evaluated through prediction.

5. CONCLUSION

This paper turns a convention into a criterion. The sampling optimizer never needed the convexity that made linear Koopman rollouts the convention, so the question is no longer which lift is tractable but which lift the system requires. A single testable property answers it, the variation v_Φ of the input gain $\mathbf{\Phi}(\boldsymbol{x})$ over the operating region. When that variation is large, the coupling is irreducible, no linear lift, however deep, can represent it, and bilinear becomes necessary. When it is small, the two coincide and the linear choice is sufficient. Bilinear therefore remains a safe *default* in both regimes, though in the second it gains no accuracy and needs the zero initialization of Sec. 4.6 on input-rich systems.

In closed loop the experiments return the same reading. Only the bilinear rollout parks the wheeled vehicle and holds the drone's reference at the oracle's level. This velocity-commanded regime, where the criterion bars linear lifts, is exactly the one prior linear-Koopman studies left untested. Their force- and acceleration-driven platforms sit in the low-variation regime, which is why their linear lifts succeed.

Two limitations bound the claim. The stability guarantee assumes a certified error bound that the deep dictionary does not yet provide, so the fitted constants of Sec. 4 remain an empirical proxy, and certified kernel- or SafEDMD-based identification with validation under real sensing and actuation noise is the next step. The criterion assumes that the tracked output couples to the input through smooth kinematics, so contact-rich hybrid dynamics, such as joint-level legged locomotion, fall outside its scope even when the navigation-level command satisfies the assumption. The result is a design rule with a known boundary. In sampling-based Koopman control, prefer a bilinear rollout whenever the input reaches the tracked output through a configuration-dependent gain that varies appreciably over the operating region.

DECLARATIONS

Conflict of Interest

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

Authors' Contributions

Kangmin Lee contributed to the implementation, conducted the experiments, and participated in writing the manuscript. Sanghyun Kim led the research direction and contributed to the writing of the manuscript.

Funding

This work was supported by the Korea Planning & Evaluation Institute of Industrial Technology (KEIT) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. RS-2025-11082970), by the Industrial Strategic Technology Development Program funded by MOTIE, Korea (No. RS-2025-25449039), and by the Korea Basic Science Institute (National Research Facilities and Equipment Center) grant funded by the Ministry of Science and ICT (No. RS-2025-00564593).

REFERENCES

[1] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic MPC for model-based reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 1714–1721.

[2] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *J. Guid. Control Dyn.*, vol. 40, no. 2, pp. 344–357, 2017.

[3] C. Pezzato, C. Salmi, E. Trevisan, M. Spahn, J. Alonso-Mora, and C. Hernández Corbato, "Sampling-based model predictive control leveraging parallelizable physics simulations," *IEEE Robot. Autom. Lett.*, vol. 10, no. 3, pp. 2750–2757, 2025.

[4] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proc. Natl. Acad. Sci.*, vol. 17, no. 5, pp. 315–318, 1931.

[5] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dyn.*, vol. 41, pp. 309–325, 2005.

[6] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *J. Nonlinear Sci.*, vol. 25, no. 6, pp. 1307–1346, 2015.

[7] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM J. Appl. Dyn. Syst.*, vol. 15, no. 1, pp. 142–161, 2016.

[8] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nat. Commun.*, vol. 9, p. 4950, 2018.

[9] W. Hao, Y. Fang, Z. Lu, and S. Mou, "Accelerating sampling-based control via learned linear Koopman dynamics," arXiv:2603.05385, 2026.

[10] D. Bruder, X. Fu, and R. Vasudevan, "Advantages of bilinear Koopman realizations for the modeling and control of systems with unknown dynamics," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4369–4376, 2021.

[11] C. Folkestad and J. W. Burdick, "Koopman NMPC: Koopman-based learning and nonlinear model predictive control of control-affine systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 7350–7356.

[12] R. Strässer, K. Worthmann, I. Mezić, J. Berberich, M. Schaller, and F. Allgöwer, “An overview of Koopman-based control: From error bounds to closed-loop guarantees,” *Annual Reviews in Control*, vol. 61, p. 101035, 2026.

[13] R. Strässer, M. Schaller, K. Worthmann, J. Berberich, and F. Allgöwer, “SafEDMD: A Koopman-based data-driven controller design framework for nonlinear dynamical systems,” *Automatica*, vol. 185, p. 112732, 2026.

[14] M. Korda and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” *Automatica*, vol. 93, pp. 149–160, 2018.

[15] D.-H. Nam and H.-G. Kim, “MPPI-based integrated task assignment, path planning, and control for unmanned ground vehicles,” *Int. J. Control Autom. Syst.*, vol. 22, no. 12, pp. 3641–3652, 2024.

[16] S. Lee and S. Kim, “Manifold-constrained MPPI: Real-time sampling-based control under hard constraints,” *arXiv preprint arXiv:2605.24813*, 2026.

[17] G. Williams, B. Goldfain, P. Drews, K. Saigol, J. M. Rehg, and E. A. Theodorou, “Robust sampling based model predictive control with sparse objective information,” in *Robotics: Science and Systems (RSS)*, 2018.

[18] J. S. Kim, Y. S. Quan, and C. C. Chung, “Koopman operator-based model identification and control for automated driving vehicle,” *Int. J. Control Autom. Syst.*, vol. 21, no. 8, pp. 2431–2443, 2023.

[19] L. C. Iacob, R. Tóth, and M. Schoukens, “Koopman form of nonlinear systems with inputs,” *Automatica*, vol. 162, p. 111525, 2024.

[20] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A physics engine for model-based control,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012, pp. 5026–5033.

[21] M. Kasaei, H. Kasaei, and M. Khadem, “SoftManiSim: A fast simulation framework for multi-segment continuum manipulators tailored for robot learning,” in *Proc. Conf. Robot Learning (CoRL)*, 2024.

[22] L. Han, K. Peng, W. Chen, and Z. Liu, “A data-driven Koopman modeling framework with application to soft robots,” *Int. J. Control Autom. Syst.*, vol. 23, no. 1, pp. 249–261, 2025.

Kangmin Lee received his B.S. degree in Mechanical Engineering from Kyung Hee University, South Korea, in 2026, where he is currently pursuing the Ph.D. degree. His research interests include optimal control, model predictive control, and reinforcement learning.

Sanghyun Kim received the B.S. and Ph.D. degrees from Seoul National University, South Korea, in 2012 and 2020, respectively. From 2020 to 2023, he was a Senior Researcher at the Korea Institute of Machinery and Materials (KIMM), South Korea. He is currently an Assistant Professor in the Department of Mechanical Engineering at Kyung Hee University. His research interests include control and planning for redundant robotic systems.