Spectrally Normalized Koopman Lifting for Provably Stable First-Order Nonlinear Optimal Control

Jong-Han Kim * **Jiwoo Choi** * **Jibon Kim** *

* *Department of Aerospace Engineering and Program in Aerospace Systems Convergence at Inha University, Incheon, South Korea (e-mails: jonghank@inha.ac.kr, {jiwoochoi, jibonkim}@inha.edu)*

Abstract: This paper presents *i-LiftProj*, a first-order nonlinear optimal control framework based on spectrally normalized invertible Koopman lifting. An augmented i-ResNet replaces the unconstrained LiftProj autoencoder, yielding a bi-Lipschitz ambient lifting map with an explicit condition-number bound. A fixed-point inverse removes learned-decoder reconstruction error from the projection step. Because the lifted linear model remains approximate, we analyze ADMM through the surrogate dynamics manifold induced by the learned lifting and derive an asymptotic residual-floor bound controlled by the lifting condition number and Koopman-model mismatch.

Keywords: Nonlinear Optimal Control, Koopman Operator, Invertible Neural Networks, Alternating Direction Method of Multipliers, Convergence Analysis.

1. INTRODUCTION

Nonlinear optimal control is central to modern autonomous systems, including agile robotics and aerospace guidance (Blackmore, 2016). The main computational difficulty is solving finite-horizon optimization problems subject to nonlinear dynamics $x_{t+1} = f(x_t, u_t)$ in real time. Sequential Convex Programming (SCP) is a standard approach (Szmuk et al., 2017), but repeated linearization and matrix factorization can be expensive for high-dimensional problems.

First-order splitting methods, especially ADMM (Boyd et al., 2011), provide a scalable alternative by separating convex objectives and constraints from nonlinear dynamics. Their main bottleneck is the projection onto the nonlinear dynamics manifold; computing this projection exactly is itself a nonlinear program and may be comparable in difficulty to the original problem (Choi et al., 2026; Kim et al., 2026).

LiftProj (Choi and Kim, 2025) addresses this bottleneck by learning a Koopman-type lifting in which the nonlinear dynamics are approximated by a lifted linear model. The dynamics projection is then replaced by lifting, closed-form projection onto a linear dynamics subspace, and retraction to the physical state space. However, the original autoencoder parameterization does not control the Lipschitz constants of the encoder and decoder. Consequently, small lifted-space corrections may be amplified in the physical space, producing projection drift or Lipschitz explosion (Gouk et al., 2021); this conflicts with bounded-defect requirements used in nonconvex ADMM analyses (Hong et al., 2016; Wang et al., 2019).

⋆ This work was supported in part by the Space Challenge Program funded by KASA under Grant No. RS-2025-16063807 and in part by the Theater Defense Research Center funded by DAPA under Grant No. UD240002SD. (Corresponding author: Jong-Han Kim.)

This paper proposes *i-LiftProj*, a spectrally normalized invertible Koopman lifting framework for first-order nonlinear optimal control. The central design goal is not only to learn a lifted linear dynamics model, but also to make the induced projection operator geometrically well conditioned.

The contributions are as follows. First, we replace the unconstrained LiftProj autoencoder with an augmented invertible ResNet whose residual branch is made contractive by spectral normalization, yielding an explicit bi-Lipschitz condition-number bound for the ambient lifting. Second, we replace the learned decoder with fixed-point inversion of the learned ambient map, eliminating decoder approximation error from the projection step. Third, since the lifted linear dynamics are approximate, we analyze the method through the learned surrogate dynamics manifold rather than identifying it with the true nonlinear dynamics manifold. The resulting residual-floor bound separates architectural conditioning from Koopman-model mismatch.

2. PRELIMINARIES

2.1 Problem Setting

We consider the discrete-time finite-horizon nonlinear optimal control problem

$$\begin{aligned} \underset{x,u}{\text{minimize}} \quad & l(x_1, \dots, x_N, u_0, \dots, u_{N-1}) \\ \text{subject to} \quad & x_{t+1} = f(x_t, u_t), \quad x_0 = x_{\text{init}}, \\ & (x_t, u_t) \in \mathcal{C}^{\text{con}}, \end{aligned} \tag{1}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, f is the discretized nonlinear dynamics, and $\mathcal{C}^{\text{con}}$ denotes convex path constraints. Let $y = (x, u)$ with $x = (x_1, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$. The nonlinear dynamics manifold is

$$\mathcal{C}^{\text{dyn}} := \{(x, u) \mid x_{t+1} = f(x_t, u_t),\ t = 0, \dots, N-1\}.$$

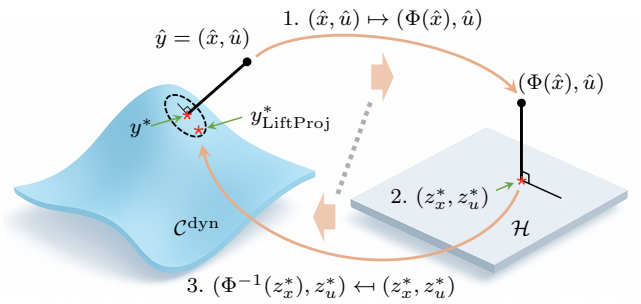

Fig. 1. The LiftProj operator $\mathcal{T}_\Phi := \Phi^{-1} \circ \Pi_{\mathcal{H}} \circ \Phi$.

2.2 Koopman Lifting and LiftProj

Koopman theory represents nonlinear dynamics through the evolution of observables. For a discrete-time system without input, the Koopman operator advances an observable ψ by $(\mathcal{K}\psi)(x_t) = \psi(f(x_t))$. For control synthesis, one seeks a finite-dimensional vector of observables $\Phi(x) \in \mathbb{R}^{N_L}$ such that the controlled dynamics are approximately linear in the lifted space:

$$\Phi(x_{t+1}) \approx A\Phi(x_t) + Bu_t.$$

Although exact finite-dimensional invariant subspaces rarely exist for general nonlinear systems, a learned finite-dimensional lifting can still provide a useful surrogate model for optimization.

LiftProj (Choi and Kim, 2025) exploits this surrogate model inside ADMM. Define the lifted linear dynamics subspace

$$\mathcal{H} := \{(z,u) \mid z_{t+1} = Az_t + Bu_t,\ t = 0,\dots,N-1\}.$$

Given an intermediate point $\hat{y} = (\hat{x},\hat{u})$, LiftProj replaces the nonlinear projection onto $\mathcal{C}^{\text{dyn}}$ by the lifted retraction

$$\mathcal{T}_\Phi := \Phi^{-1} \circ \Pi_{\mathcal{H}} \circ \Phi, \tag{2}$$

where the input coordinates are passed through unchanged. Operationally, the trajectory is embedded into the lifted space, projected onto the linear dynamics subspace $\mathcal{H}$, and retracted back to the physical state space; see Fig. 1. The middle step is a linear least-squares projection and is therefore computationally inexpensive compared with solving a nonlinear projection problem.

The accuracy and stability of this approximate projection depend on the conditioning of the lifting map. If Φ is bi-Lipschitz, projection errors in the lifted space are amplified in the physical space by at most the condition number

$$\kappa(\Phi) := L_\Phi L_{\Phi^{-1}}.$$

The original LiftProj parameterized Φ and Φ^{-1} by standard neural networks, so $\kappa(\Phi)$ was not controlled a priori. Consequently, a small lifted-space correction may become a large physical-space correction, a phenomenon often associated with Lipschitz explosion (Gouk et al., 2021). This motivates replacing the autoencoder with an invertible architecture whose Lipschitz constants are explicitly bounded.

3. PROPOSED METHOD: I-LIFTPROJ

3.1 Spectrally Normalized Augmented i-ResNet

Invertible residual networks are dimension preserving, whereas Koopman lifting generally requires $N_L > n$. We augment the state by zero padding,

$$\tilde{x} = (x, 0_{N_L-n}) \in \mathbb{R}^{N_L},$$

and define the ambient residual lifting

$$z = \Phi(\tilde{x}) = \tilde{x} + g_\theta(\tilde{x}). \tag{3}$$

State augmentation resolves the dimension mismatch while preserving invertibility (Dupont et al., 2019).

To make g_θ contractive by construction, each affine layer is spectrally scaled. For $g_\theta = \bar{W}_L \circ \varphi_{L-1} \circ \bar{W}_{L-1} \circ \cdots \circ \varphi_1 \circ \bar{W}_1$, set

$$\bar{W}_\ell = c_\ell W_\ell / \sigma_{\max}(W_\ell), \qquad 0 < c_\ell < 1. \tag{4}$$

If φ_ℓ is L_{φ_ℓ}-Lipschitz and

$$\prod_{\ell=1}^{L} c_\ell \prod_{\ell=1}^{L-1} L_{\varphi_\ell} \le c < 1,$$

then $L_{g_\theta} \le c$. Hence $\Phi = I + g_\theta$ is a bi-Lipschitz diffeomorphism by the Banach fixed-point theorem (Behrmann et al., 2019), and we have $(1-c)\,\|u-v\| \le \|\Phi(u) - \Phi(v)\| \le (1+c)\,\|u-v\|$. Therefore

$$\kappa(\Phi) \le (1+c)/(1-c). \tag{5}$$

Thus c acts as a stability–expressivity knob: smaller c improves conditioning and inversion rate, while larger c increases representational flexibility.

3.2 Decoder-Free Retraction

The original LiftProj framework used a separately trained decoder to approximate Φ^{-1}. In contrast, i-LiftProj computes the inverse of the learned ambient diffeomorphism directly. Given a projected lifted point z^*, the inverse is obtained by the fixed-point iteration

$$\tilde{x}^{(j+1)} = z^* - g_\theta(\tilde{x}^{(j)}), \qquad \tilde{x}^{(0)} = z^*. \tag{6}$$

Since $L_{g_\theta} \le c < 1$, the map $\tilde{x} \mapsto z^* - g_\theta(\tilde{x})$ is a contraction. Therefore, (6) converges linearly to the unique inverse $\Phi^{-1}(z^*)$ by the Banach fixed-point theorem.

The contraction factor also gives a direct iteration-complexity estimate. Let $\tilde{x}^*$ denote the exact inverse and define, for a trajectory,

$$\Delta_{\text{fpi}} := \max_{t=0,\dots,N} \|\tilde{x}_t^{(1)} - \tilde{x}_t^{(0)}\|.$$

Then each time node satisfies

$$\|\tilde{x}_t^{(j)} - \tilde{x}_t^*\| \le \tfrac{c^j}{1-c}\|\tilde{x}_t^{(1)} - \tilde{x}_t^{(0)}\| \le \tfrac{c^j}{1-c}\Delta_{\text{fpi}}.$$

Thus a sufficient number of inverse iterations for ε_{fpi} is

$$N_{\text{fpi}}(\varepsilon_{\text{fpi}}) = \left\lceil \left[\frac{\log(\Delta_{\text{fpi}}/((1-c)\varepsilon_{\text{fpi}}))}{\log(1/c)} \right]_+ \right\rceil. \tag{7}$$

The per-ADMM overhead is therefore $\mathcal{O}(NC_gN_{\text{fpi}})$, where C_g is the cost of one residual-network evaluation. This overhead is linear in the horizon length and parallelizable over time nodes.

After inversion, the physical state is recovered by slicing,

$$x^* = [\tilde{x}^*]_{1:n}.$$

The remaining components $r = [\tilde{x}^*]_{n+1:N_L}$ form the augmentation residual. If the projected lifted point lies exactly on the image of the zero-padded state manifold, then $r = 0$. In practice, the projection onto $\mathcal{H}$ may perturb the lifted point slightly off this image. Let $P_{\mathcal{S}}$ denote the projection onto $\mathcal{S} = \mathbb{R}^n \times \{0\}^{N_L-n}$. The implemented retraction is therefore $\bar{\mathcal{T}}_\Phi = P_{\mathcal{S}} \circ \mathcal{T}_\Phi$, and if $\|P_{\mathcal{S}}\mathcal{T}_\Phi(v) - \mathcal{T}_\Phi(v)\| \le \varepsilon_r$, the slicing step contributes an additional bounded perturbation ε_r to the augmented retraction analyzed in Section 4.

3.3 Training Objective

Let the physical lifting induced by the augmented map be $\phi(x) := \Phi(x, 0_{N_L-n})$. For a data pair (x_t, u_t) with $x_{t+1} = f_{\mathrm{RK}}(x_t, u_t)$, define

$$r_{\mathrm{lin}}(x_t, u_t) := \phi(f_{\mathrm{RK}}(x_t, u_t)) - (A\phi(x_t) + Bu_t).$$

The proposed training loss is

$$\mathcal{L}_{\text{i-LiftProj}} = \sum_{t=0}^{N-1} \Big(\overbrace{\|r_{\mathrm{lin}}(x_t, u_t)\|^2}^{\mathcal{L}^{\mathrm{DD}}_{\mathrm{lin},t}} + \eta \overbrace{\big\|D_{(x,u)} r_{\mathrm{lin}}(x_t, u_t)\big\|_F^2}^{\mathcal{L}^{\mathrm{PI}}_{\mathrm{lin},t}} + \gamma \underbrace{\big\|D_x\phi(x_t)^\top D_x\phi(x_t) - I\big\|_{1,1}}_{\mathcal{L}_{\mathrm{iso},t}} \Big). \quad (8)$$

The first term fits one-step lifted dynamics, the second suppresses sensitivity of the lifted residual, and the last promotes near-isometry on the sampled physical-state manifold. Unlike LiftProj, no autoencoder reconstruction loss is needed because inversion is algorithmic.

The Jacobian residual term is included because small pointwise one-step error alone does not guarantee reliable behavior under ADMM perturbations. During projection, the iterates may move away from the sampled trajectory points, and the local sensitivity of r_{lin} determines how quickly lifted-model mismatch grows. The near-isometry term complements spectral normalization by improving conditioning specifically on the physical state manifold used for control.

3.4 Relation to LiftProj

The loss landscape of i-LiftProj is simpler than that of the original LiftProj. LiftProj used linearization, autoencoding, and physical prediction losses in both data-driven and physics-informed forms:

$$\mathcal{L}_{\mathrm{LiftProj}} = \sum\nolimits_{\star \in \{\mathrm{lin,AE,dyn}\}} \left(w^{\mathrm{DD}}_\star \mathcal{L}^{\mathrm{DD}}_\star + w^{\mathrm{PI}}_\star \mathcal{L}^{\mathrm{PI}}_\star \right).$$

In i-LiftProj, the autoencoding losses are unnecessary because the inverse is computed algorithmically rather than learned. The physical prediction losses are also not imposed separately; instead, the effect of lifted-model error is tracked in the convergence analysis through the Koopman-model mismatch ε_K. Consequently, the proposed objective focuses on three roles: fitting the lifted linear dynamics, suppressing the sensitivity of the lifted residual, and regularizing the local geometry of the lifting map. Spectral normalization controls global conditioning, while the near-isometry term in (8) promotes well-conditioned behavior on the sampled physical-state manifold.

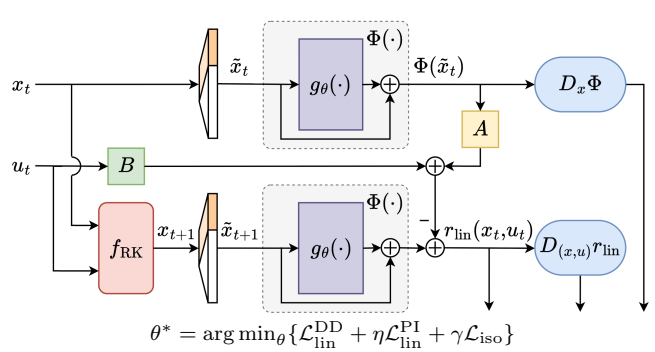

Fig. 2. i-LiftProj architecture and training losses.

4. CONVERGENCE ANALYSIS

The lifted model is approximate, so the image of the true dynamics manifold need not coincide with $\mathcal{H}$. We therefore analyze ADMM with respect to the surrogate dynamics manifold induced by the learned lifting and track the mismatch to the true manifold explicitly. In this section, $y = (\tilde{x}, u)$ denotes the augmented trajectory variable, and

$$\tilde{\Phi}(y) := (\Phi(\tilde{x}_0), \ldots, \Phi(\tilde{x}_N), u), \qquad \kappa(\Phi) := L_{\tilde{\Phi}} L_{\tilde{\Phi}^{-1}}.$$

Define

$$\hat{\mathcal{C}}^{\mathrm{dyn}} := \{y \in \mathcal{D} \mid \tilde{\Phi}(y) \in \mathcal{H}\}, \qquad \mathcal{T}_\Phi := \tilde{\Phi}^{-1} \circ \Pi_{\mathcal{H}} \circ \tilde{\Phi}, \quad (9)$$

where $\mathcal{D}$ is a compact local operating region containing the iterates. The local Koopman-model mismatch is

$$\varepsilon_K := \sup_{y \in \mathcal{C}^{\mathrm{dyn}} \cap \mathcal{D}} \mathrm{dist}(\tilde{\Phi}(y), \mathcal{H}). \quad (10)$$

This distinction between $\mathcal{C}^{\mathrm{dyn}}$ and $\hat{\mathcal{C}}^{\mathrm{dyn}}$ is essential. Even if the projection onto $\mathcal{H}$ is computed exactly, the resulting retraction is guaranteed to land on the learned surrogate manifold, not necessarily on the true nonlinear dynamics manifold. The mismatch constant ε_K measures the worst local violation of the lifted linear dynamics by true dynamically feasible trajectories. Hence ε_K is not a numerical projection error; it is a modeling error induced by the finite-dimensional learned Koopman approximation. The analysis below separates this modeling error from the geometric distortion caused by the lifting map.

The analyzed consensus problem is

$$\begin{aligned} \underset{y,z}{\text{minimize}} \quad & F(y) + I_{\hat{\mathcal{C}}^{\mathrm{dyn}}}(z) \\ \text{subject to} \quad & y = z, \end{aligned}$$

with scaled augmented Lagrangian

$$\hat{\mathcal{L}}_\rho = F(y) + I_{\hat{\mathcal{C}}^{\mathrm{dyn}}}(z) + \frac{\rho}{2}\|y - z + \lambda\|^2 - \frac{\rho}{2}\|\lambda\|^2.$$

The i-LiftProj ADMM iteration is

$$\begin{aligned} z^{(k+1)} &= \mathcal{T}_\Phi(y^{(k)} + \lambda^{(k)}), \\ y^{(k+1)} &= \arg\min_y \hat{\mathcal{L}}_\rho(y, z^{(k+1)}, \lambda^{(k)}), \\ \lambda^{(k+1)} &= \lambda^{(k)} + y^{(k+1)} - z^{(k+1)}. \end{aligned} \quad (11)$$

Lemma 1. (Surrogate-Manifold Distortion Bound). For any $v \in \mathcal{D}$, the augmented i-LiftProj retraction satisfies

$$\|\mathcal{T}_\Phi(v) - v\| \le \kappa(\Phi)\,\mathrm{dist}(v, \hat{\mathcal{C}}^{\mathrm{dyn}}), \quad (12)$$

$$\|\mathcal{T}_\Phi(v) - v\| \le \kappa(\Phi)\,\mathrm{dist}(v, \mathcal{C}^{\mathrm{dyn}} \cap \mathcal{D}) + L_{\tilde{\Phi}^{-1}}\varepsilon_K. \quad (13)$$

Moreover, $\mathrm{dist}(v, \hat{\mathcal{C}}^{\mathrm{dyn}}) \le \mathrm{dist}(v, \mathcal{C}^{\mathrm{dyn}} \cap \mathcal{D}) + L_{\tilde{\Phi}^{-1}}\varepsilon_K$.

Proof. (Sketch). Since $\mathcal{T}_\Phi(v) = \tilde{\Phi}^{-1}(\Pi_{\mathcal{H}}\tilde{\Phi}(v))$, we have $\|\mathcal{T}_\Phi(v) - v\| \le L_{\tilde{\Phi}^{-1}}\,\mathrm{dist}(\tilde{\Phi}(v), \mathcal{H})$. If $w \in \hat{\mathcal{C}}^{\mathrm{dyn}}$, then $\tilde{\Phi}(w) \in \mathcal{H}$, giving (12) after taking the infimum over w. Inserting any $y \in \mathcal{C}^{\mathrm{dyn}} \cap \mathcal{D}$ and using the definition of ε_K gives (13); the final distance bound follows by applying the same estimate at a nearest true-manifold point. □

For the sliced implementation $\bar{\mathcal{T}}_\Phi := P_{\mathcal{S}} \circ \mathcal{T}_\Phi$, Lemma 1 gains an additive perturbation. Specifically, if the slicing residual is bounded by $\|P_{\mathcal{S}}\mathcal{T}_\Phi(v) - \mathcal{T}_\Phi(v)\| \le \varepsilon_r$, then

$$\|\bar{\mathcal{T}}_\Phi(v) - v\| \le \kappa(\Phi)\,\mathrm{dist}(v, \hat{\mathcal{C}}^{\mathrm{dyn}}) + \varepsilon_r.$$

Thus the ideal augmented analysis remains applicable to the implemented method as long as the augmentation residual is monitored or bounded in the operating region.

Theorem 2. (Residual-Floor Bound under Surrogate Mismatch). Assume F is L_F-smooth and μ_F-strongly convex, and $\hat{\mathcal{L}}_\rho$ is bounded from below. Let $w^{(k)} = (y^{(k)}, z^{(k)}, \lambda^{(k)})$ be generated by (11), and set $v^{(k)} := y^{(k)} + \lambda^{(k)}$. Assume $v^{(k)}, \mathcal{T}_\Phi(v^{(k)}) \in \mathcal{D}$ and $\operatorname{dist}(v^{(k)}, \mathcal{C}^{\mathrm{dyn}} \cap \mathcal{D}) \le \delta$ for all k. Define

$$\hat{\delta} := \delta + L_{\tilde{\Phi}^{-1}} \varepsilon_K, \qquad \beta := \frac{\mu_F + \rho}{2} - \frac{L_F^2}{\rho}.$$

If ρ is large enough that $\beta > 0$, then

$$\liminf_{k\to\infty} \|y^{(k+1)} - y^{(k)}\|^2 \le \frac{\rho\hat{\delta}^2}{2\beta}\big(\kappa(\Phi)^2 - 1\big), \tag{14}$$

$$\liminf_{k\to\infty} \|y^{(k+1)} - z^{(k+1)}\|^2 \le \frac{L_F^2\hat{\delta}^2}{2\rho\beta}\big(\kappa(\Phi)^2 - 1\big). \tag{15}$$

Proof. (Sketch). Let $\hat{z}_*^{(k+1)} = \Pi_{\hat{\mathcal{C}}^{\mathrm{dyn}}}(v^{(k)})$ and define

$$\eta_k := \frac{\rho}{2}\Big(\|\mathcal{T}_\Phi(v^{(k)}) - v^{(k)}\|^2 - \operatorname{dist}(v^{(k)}, \hat{\mathcal{C}}^{\mathrm{dyn}})^2\Big).$$

Lemma 1 implies $\eta_k \le \bar{\eta} := \rho\hat{\delta}^2(\kappa(\Phi)^2 - 1)/2$. The exact y-update gives a decrease of

$$(\mu_F + \rho)\|y^{(k+1)} - y^{(k)}\|^2/2.$$

From the y-optimality condition and the dual update, we have $\rho\lambda^{(k+1)} = -\nabla F(y^{(k+1)})$, so

$$\|\lambda^{(k+1)} - \lambda^{(k)}\| \le (L_F/\rho)\|y^{(k+1)} - y^{(k)}\|,$$

and the dual update step increases $\hat{\mathcal{L}}_\rho$ by at most $(L_F^2/\rho)\|y^{(k+1)} - y^{(k)}\|^2$. Therefore

$$\hat{\mathcal{L}}_\rho(w^{(k+1)}) - \hat{\mathcal{L}}_\rho(w^{(k)}) \le -\beta\|y^{(k+1)} - y^{(k)}\|^2 + \bar{\eta}.$$

Summing this inequality and using lower boundedness gives (14). Since $y^{(k+1)} - z^{(k+1)} = \lambda^{(k+1)} - \lambda^{(k)}$, the dual-increment bound gives (15). □

Theorem 2 should be read as a residual-floor result for the learned surrogate problem, not as exact feasibility with respect to $\mathcal{C}^{\mathrm{dyn}}$. It separates architectural conditioning from model mismatch: $\kappa(\Phi)^2 - 1$ is controlled by spectral normalization, while $\hat{\delta} = \delta + L_{\tilde{\Phi}^{-1}}\varepsilon_K$ captures the local true-manifold distance and lifted surrogate error; the sliced implementation adds the augmentation residual as a bounded perturbation. Using (5),

$$\kappa(\Phi)^2 - 1 \le \Big(\frac{1+c}{1-c}\Big)^2 - 1 = \frac{4c}{(1-c)^2}.$$

Thus the spectral contraction level c directly upper-bounds the architecture-induced part of the residual floor. Smaller c improves the residual-floor constant and accelerates fixed-point inversion, but may reduce the expressivity of the lifted model and increase ε_K. Conversely, larger c can improve the fit of the lifted linear model but weakens conditioning. This makes c a practical stability–expressivity knob.

For the sliced implementation, the same argument applies with an enlarged inexactness term. If the uniform slicing residual satisfies $\big\|P_{\mathcal{S}}\mathcal{T}_\Phi(v^{(k)}) - \mathcal{T}_\Phi(v^{(k)})\big\| \le \varepsilon_r$, then the z-step error can be bounded by

$$\bar{\eta}_r = \frac{\rho}{2}\Big[\big(\kappa(\Phi)^2 - 1\big)\hat{\delta}^2 + 2\kappa(\Phi)\hat{\delta}\varepsilon_r + \varepsilon_r^2\Big].$$

Hence the residual floor is enlarged continuously with the measured augmentation residual. In the ideal case $\varepsilon_r = 0$, this reduces to the bound in Theorem 2.

5. CONCLUDING REMARKS

This paper presented *i-LiftProj*, a first-order nonlinear optimal control framework based on spectrally normalized invertible Koopman lifting. By replacing the unconstrained LiftProj autoencoder with an augmented i-ResNet, the method provides an explicit bi-Lipschitz condition-number bound for the ambient lifting map. The decoder-free fixed-point inverse removes learned-decoder reconstruction error and gives a direct tradeoff between inversion cost and the contraction level.

The analysis interprets the ADMM projection through the learned surrogate dynamics manifold rather than the exact nonlinear dynamics manifold. This yields an asymptotic residual-floor bound controlled by the lifting condition number and the Koopman-model mismatch. The result clarifies the roles of the spectral contraction parameter, the lifted linearization error, and the augmentation residual in determining the stability of the first-order optimization loop.

REFERENCES

Behrmann, J., Grathwohl, W., Chen, R.T., Duvenaud, D., and Jacobsen, J.H. (2019). Invertible residual networks. In *Proc. Int. Conf. Mach. Learn. (ICML)*, 573–582. PMLR.

Blackmore, L. (2016). Autonomous precision landing of space rockets. In *Frontiers of Eng.: Rep. on Leading-Edge Eng. from the 2016 Symp.*, volume 46, 15–20. The Bridge Washington, DC, USA.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1), 1–122.

Choi, J. and Kim, J.H. (2025). LiftProj: Physics-informed Koopman lifting and projection for nonlinear optimal control via first-order optimization. *IEEE Control Syst. Lett.*, 9, 817–822. Also presented at the 64th IEEE Conf. Decis. Control (CDC), 2025.

Choi, J., Lee, D., and Kim, J.H. (2026). Neural projection operators for real-time 6-dof powered descent guidance. In *Proc. AIAA SCITECH 2026 Forum*, 1169.

Dupont, E., Doucet, A., and Teh, Y.W. (2019). Augmented neural ODEs. *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 32.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M.J. (2021). Regularisation of neural networks by enforcing Lipschitz continuity. *Mach. Learn.*, 110(2), 393–416.

Hong, M., Luo, Z.Q., and Razaviyayn, M. (2016). Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM J. Optim.*, 26(1), 337–364.

Kim, Y.J., Choi, J., Choi, J., and Kim, J.H. (2026). A first-order approach for nonlinear optimal control under nonconvex constraints. *Optim. Eng.*, 27.

Szmuk, M., Eren, U., and Açıkmeşe, B. (2017). Successive convexification for mars 6-dof powered descent landing guidance. In *Proc. AIAA Guid. Navig. Control Conf.*, 1500.

Wang, Y., Yin, W., and Zeng, J. (2019). Global convergence of ADMM in nonconvex nonsmooth optimization. *J. Sci. Comput.*, 78(1), 29–63.