i-LiftProj: Spectrally Normalized Invertible Koopman Lifting for First-Order Nonlinear Optimal Control

Jong-Han Kim[1,†] Jiwoo Choi[1] Jibon Kim[1]

Abstract

This paper presents *i-LiftProj*, a first-order optimization framework for nonlinear optimal control based on a spectrally normalized invertible Koopman lifting. Building on the recently proposed *LiftProj* framework, which performs ADMM dynamics projections in a lifted linear space, we replace the unconstrained autoencoder with an augmented Invertible ResNet whose ambient lifting map is bi-Lipschitz by construction. We also replace the learned decoder with a fixed-point inverse of the learned ambient diffeomorphism, thereby removing learned-decoder approximation error from the projection step. Since the lifted linear model remains approximate, we analyze i-LiftProj through a surrogate dynamics manifold induced by the learned lifting rather than identifying it with the true nonlinear dynamics manifold. Under standard smoothness and strong-convexity assumptions on the convex ADMM subproblem, we derive an asymptotic residual-floor bound whose size is explicitly controlled by the lifting condition number and by the mismatch between the true dynamics manifold and its lifted linear surrogate. Numerical experiments on a 6-DoF rocket powered descent guidance problem show that i-LiftProj preserves solution quality and constraint satisfaction while yielding smoother residual reduction than the LiftProj baseline.

1 Introduction

Nonlinear optimal control is a cornerstone of modern autonomy, from agile robotics to aerospace guidance [1]. The core challenge lies in solving non-convex optimization problems subject to complex dynamics, $x_{t+1}=f(x_t, u_t)$, in real-time. While Sequential Convex Programming (SCP) [2] is the industry standard, it relies on iterative linearization and computationally expensive matrix factorizations, limiting its frequency in high-dimensional state spaces.

First-order methods, particularly the Alternating Direction Method of Multipliers (ADMM), offer a scalable alternative by decomposing the problem into simpler subproblems [3]. In this framework, the consensus constraints are split between convex objectives (costs, state limits) and the nonconvex dynamics. Consequently, the convergence of the algorithm hinges entirely on the efficient evaluation of the *projection operator* onto the dynamics manifold. Computing this problem exactly is itself a non-convex nonlinear programming (NLP) problem, often as computationally demanding as the original control problem [4, 5].

To bypass this bottleneck, the *LiftProj* framework [6] proposed a novel geometric approach: leveraging Koopman Operator Theory to lift the dynamics into a latent space where the manifold becomes a linear subspace. In this lifted space, the intractable projection reduces to a simple closed-form linear least squares problem.

Despite its promise, the original LiftProj framework suffers from a critical theoretical gap: stability. The convergence of nonconvex ADMM is predicated on the projection operator being non-expansive (or having bounded distortion). Standard deep autoencoders used to approximate the Koopman operator provide no guarantees on their Lipschitz constants. In practice, this leads to *Lipschitz explosion*, where small corrections in the latent space map to catastrophic jumps in the physical state space, causing the solver to diverge [7].

We propose *i-LiftProj*, a spectrally normalized Koopman lifting framework with bi-Lipschitz stability by construction. Main contributions are as follows:

1. *Bi-Lipschitz Stability by Construction:* We implement an augmented invertible ResNet (i-ResNet) with spectral normalization, showing that the ambient lifting map is a bi-Lipschitz diffeomorphism on the augmented space with a bounded condition number.
2. *Algorithmic Inversion:* We replace the learned decoder with a decoder-free fixed-point inversion procedure, removing learned-decoder reconstruction error.
3. *Residual-Floor Analysis:* We analyze the lifted ADMM scheme via a surrogate dynamics manifold induced by the learned lifting, and derive an asymptotic residual-floor bound controlled by the lifting condition number and the Koopman-model mismatch.

2 Preliminaries

2.1 Nonlinear Optimal Control Problem

We consider the discrete-time finite-horizon optimal control problem for a nonlinear dynamical system. The objective is to minimize a convex cost function l subject to dynamics and constraints:

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

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ denote the state and control input at time step t. The function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ repre-

[*]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).

[1]The authors are with the Department of Aerospace Engineering and the Program in Aerospace Systems Convergence at Inha University, Incheon 21999, South Korea. jonghank@inha.ac.kr; {jiwoochoi,jibonkim}@inha.edu; [†]J.-H. Kim is also with the Aerospace Systems Research Institute at Inha University.

sents the discretized nonlinear dynamics (*e.g.*, via Runge-Kutta integration). The set $\mathcal{C}^{\text{con}}$ represents convex path constraints (*e.g.*, box constraints, glide-slope cones) [6]. Here, we define the trajectory vectors $x = (x_1, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ where the vertical concatenation notation $(a, b, c) = [a^\top \; b^\top \; c^\top]^\top$ was used.

2.2 Koopman Operator Theory

Koopman theory [8] enables the global linearization of nonlinear dynamics by lifting the state into an infinite-dimensional Hilbert space of observable functions $\psi : \mathbb{R}^n \to \mathbb{R}$. The Koopman operator $\mathcal{K}$ is an infinite-dimensional linear operator that advances these observables along the flow of the system, $(\mathcal{K}\psi)(x_t) = \psi(f(x_t)) = \psi(x_{t+1})$.

For practical control synthesis, we seek a finite-dimensional invariant subspace spanned by a vector of observables $\Phi(x) = (\phi_1(x), \dots, \phi_{N_L}(x))$. In this lifted space $\mathbb{R}^{N_L}$ (where $N_L \gg n$), the dynamics are approximated by a linear evolution matrix $A \in \mathbb{R}^{N_L \times N_L}$ and input matrix $B \in \mathbb{R}^{N_L \times m}$, *i.e.*, $\Phi(x_{t+1}) \approx A\Phi(x_t) + Bu_t$. The minimization of this residual error allows linear control techniques to be applied to the nonlinear system.

2.3 Review of LiftProj

The *LiftProj* framework [6] solves (1) using ADMM by splitting the constraints into convex bounds $\mathcal{C}^{\text{con}}$ and the nonconvex dynamics manifold $\mathcal{C}^{\text{dyn}}$, defined as:

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

The core computational bottleneck in ADMM is the projection onto this manifold, denoted as $\Pi_{\mathcal{C}^{\text{dyn}}}(\hat{y})$, where $\hat{y} = (\hat{x}, \hat{u})$ is the intermediate primal variable. Exact projection requires solving an NLP problem at every iteration.

LiftProj approximates this projection efficiently via a three-step procedure in the lifted space (see Figure 1):

1) Embedding. The current trajectory estimate $\hat{y}$ is mapped to the lifted space using the encoder Φ:

$$\hat{z} = (\Phi(\hat{x}), \hat{u}), \quad \text{where} \quad \Phi(\hat{x}) = (\Phi(x_0), \dots, \Phi(x_N)).$$

2) Linear Projection. The lifted trajectory $\hat{z}$ is projected onto the linear dynamics subspace

$$\mathcal{H} := \{(z_x, u) \mid z_{t+1} = Az_t + Bu_t, \; \forall t \in [0, N-1]\}.$$

This is a convex quadratic program (QP) with a closed-form solution, $z^* = \arg\min_{z \in \mathcal{H}} \|z - \hat{z}\|^2$.

3) Retraction. The projected lifted state z_x^* is mapped back to the physical state space using the decoder Φ^{-1}:

$$x^* = \Phi^{-1}(z_x^*), \quad u^* = z_u^*.$$

While computationally efficient, the geometric validity of this approximate projection relies heavily on the conditioning of the mapping Φ. It has been established [6] that the projection error in the physical space is bounded by the distance to the manifold scaled by the bi-Lipschitz condition number, $\kappa(\Phi) = L_\Phi L_{\Phi^{-1}}$.

In the original LiftProj framework, Φ and Φ^{-1} were parameterized by standard MLPs without Lipschitz constraints. Hence, the condition number $\kappa(\Phi)$ is unbounded a priori, allowing the projection error to be arbitrarily amplified. This unconstrained *projection drift* violates the bounded defect condition required for ADMM convergence [9, 10], leading to potential divergence (often termed *Lipschitz explosion*) when the solver explores regions far from the training data. This critical instability necessitates the transition to a rigorously stable architecture.

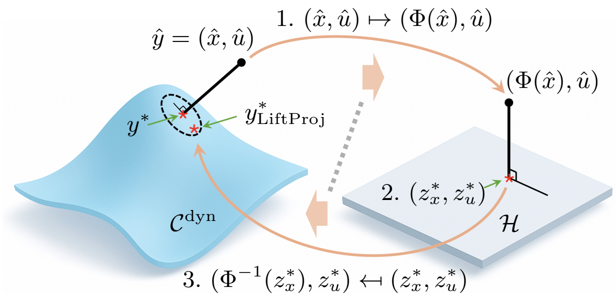


Figure 1: The LiftProj operator $\mathcal{T}_\Phi := \Phi^{-1} \circ \Pi_\mathcal{H} \circ \Phi$.

2.4 Invertible Residual Networks (i-ResNet)

To guarantee the stability of the mapping Φ, we employ Invertible Residual Networks. An i-ResNet defines the mapping as a perturbation of the identity:

$$z = \Phi(x) = x + g_\theta(x),$$

with $g_\theta(x)$ being Lipschitz continuous with constant L_{g_θ}.

The *Banach Fixed Point Theorem* guarantees that this map is a bi-Lipschitz diffeomorphism if the residual block g_θ is a strict contraction, *i.e.*, $L_{g_\theta} < 1$. Under this condition, the inverse map $x = \Phi^{-1}(z)$ exists uniquely and can be computed via the fixed-point iteration [11]:

$$x^{(j+1)} = z - g_\theta(x^{(j)}), \quad \text{with} \quad x^{(0)} = z, \tag{2}$$

where the superscript (j) denotes the iteration index.

Crucially, the Lipschitz constant of the inverse is bounded by $L_{\Phi^{-1}} \leq (1 - L_{g_\theta})^{-1}$, providing the theoretical foundation for bounding the projection error in the ADMM loop.

3 Proposed Method: i-LiftProj

The core idea of i-LiftProj is to replace the heuristic autoencoder with a mathematically constrained invertible architecture that enforces the geometry required for optimization stability.

3.1 Augmented Lifting Topology

A fundamental limitation of standard invertible neural networks is that they are dimension-preserving ($N_{out} = N_{in}$). However, Koopman Operator Theory generally necessitates lifting the state $x \in \mathbb{R}^n$ into a significantly higher-dimensional space $\mathbb{R}^{N_L}$ ($N_L \gg n$) to linearly approximate the underlying nonlinear dynamics.

To resolve this topological mismatch, we introduce *State Augmentation* [12]. We define the augmented input $\tilde{x} \in \mathbb{R}^{N_L}$ by zero-padding the physical state:

$$\tilde{x} = (x, \mathbf{0}_{N_L - n}).$$

The lifting map $\Phi : \mathbb{R}^{N_L} \to \mathbb{R}^{N_L}$ is then defined as a residual transformation on this augmented space:

$$z = \Phi(\tilde{x}) = \tilde{x} + g_\theta(\tilde{x}),$$

where $g_\theta : \mathbb{R}^{N_L} \to \mathbb{R}^{N_L}$ is a deep neural network parameterized by θ. This architecture allows the network to populate the zero-padded dimensions with the nonlinear observables required to linearize the dynamics, while strictly preserving the invertible structure.

3.2 Stability via Spectral Normalization

To guarantee the invertibility of Φ and the stability of the ADMM projection step, the map must be a bi-Lipschitz diffeomorphism. The *Banach Fixed Point Theorem* guarantees invertibility if the residual block g_θ is a strict contraction [11], *i.e.*, $L_{g_\theta} < 1$.

To enforce this constraint by construction, we apply distributed spectral scaling to each affine layer of the residual branch. Let $g_\theta = \bar{W}_L \circ \varphi_{L-1} \circ \bar{W}_{L-1} \circ \cdots \circ \varphi_1 \circ \bar{W}_1$, where φ_ℓ denotes the activation function and

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

Here, $\sigma_{\max}(W_\ell)$ denotes the largest singular value of W_ℓ, so that $\|\bar{W}_\ell\|_2 \le c_\ell$. If φ_ℓ is L_{φ_ℓ}-Lipschitz, then the residual branch satisfies $L_{g_\theta} \le \prod_{\ell=1}^{L} c_\ell \prod_{\ell=1}^{L-1} L_{\varphi_\ell}$. We choose the layerwise scaling factors $\{c_\ell\}$ such that

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

which guarantees that g_θ is a strict contraction. For 1-Lipschitz activations, this reduces to $\prod_{\ell=1}^{L} c_\ell \le c$; for example, one may choose $c_\ell = c^{1/L}$ uniformly across layers.

Since $\Phi(x) = x + g_\theta(x)$ and $L_{g_\theta} \le c < 1$, for any u, v we have $(1-c)\|u - v\| \le \|\Phi(u) - \Phi(v)\| \le (1+c)\|u - v\|$. Therefore,

$$\kappa(\Phi) = L_\Phi L_{\Phi^{-1}} \le (1+c)/(1-c). \tag{3}$$

This bound acts as the *stability knob* for the optimization. The parameter c controls a stability–expressivity tradeoff: smaller values improve conditioning and fixed-point inversion rates, while overly small values may reduce the ability of the lifted model to fit complex nonlinear dynamics. In practice, c is selected by validation together with the lifted linearization losses, and state augmentation helps retain expressivity while keeping the map well conditioned.

3.3 Decoder-Free Inversion via Fixed-Point Iteration

In previous work [6], the inverse map Φ^{-1} was represented by a separate decoder network, which introduced reconstruction error. Here we remove the learned decoder and compute the inverse of the learned ambient map algorithmically. Given a lifted point z^*, we recover $\tilde{x}^*$ by fixed-point iteration:

$$\tilde{x}^{(j+1)} = z^* - g_\theta(\tilde{x}^{(j)}), \qquad \tilde{x}^{(0)} = z^*. \tag{4}$$

Because $L_{g_\theta} < 1$, this iteration converges to the unique inverse of the learned ambient map up to the prescribed tolerance.

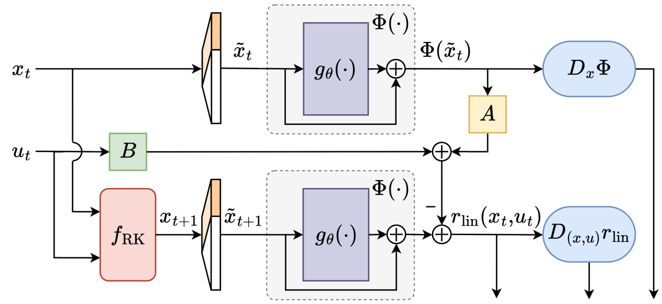
Given a target fixed-point tolerance ε_{fpi}, the number of inverse iterations can be bounded directly. For a trajectory, define $\Delta_{\text{fpi}} := \max_{t=0,\ldots,N} \|\tilde{x}_t^{(1)} - \tilde{x}_t^{(0)}\|$. Since the fixed-point map is c-contractive, each time node satisfies

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

Thus the number of iterations sufficient to guarantee the target tolerance ε_{fpi} is given by

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

and the per-ADMM overhead scales as $\mathcal{O}\left(N C_g N_{\text{fpi}}(\varepsilon_{\text{fpi}})\right)$,



$$\theta^* = \arg\min_\theta \left\{ \mathcal{L}_{\text{lin}}^{\text{DD}} + \eta \mathcal{L}_{\text{lin}}^{\text{PI}} + \gamma \mathcal{L}_{\text{iso}} \right\}$$

Figure 2: i-LiftProj architecture and loss landscape. Orange blocks represent state augmentation, and purple blocks denote the spectrally normalized residual network.

where C_g is the cost of one residual-network evaluation. The overhead is linear in the horizon length and parallelizable across time nodes. For fully connected residual branches, $C_g = \mathcal{O}(\sum_\ell d_\ell d_{\ell-1})$, so higher lifted dimensions affect the per-node network-evaluation cost, while the horizon dependence remains linear and parallelizable.

1) Recovery via Slicing. The forward map Φ is trained on augmented inputs of the form $\tilde{x} = (x, \mathbf{0}_{N_L - n})$. Hence, for lifted points that remain close to the image of the training manifold, the inverse is expected to lie near $\mathcal{S} := \mathbb{R}^n \times \{\mathbf{0}\}^{N_L - n}$. After inversion, the physical state can be obtained by extracting the first n coordinates:

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

2) Augmentation Residuals. The remaining components $r = [\tilde{x}^*]_{n+1:N_L}$ constitute the *augmentation residual*. If z^* is perfectly consistent with the system dynamics, r should be exactly zero. In practice, the ADMM linear projection step may perturb z^* off the image manifold of Φ, resulting in non-zero residuals. In implementation, we discard these components, effectively projecting $\tilde{x}^*$ orthogonally back onto the subspace $\mathcal{S}$. With $P_{\mathcal{S}}$ denoting this projection, $\|P_{\mathcal{S}}\tilde{x}^* - \tilde{x}^*\| = \|r\|$, the slicing step can be viewed as a bounded post-processing perturbation of the augmented retraction analyzed in Section 4. Theorem 2 analyzes the ideal augmented retraction; the slicing residual enters the implemented algorithm as an additional perturbation term.

3) Implicit Residual Suppression. Although we discard r, large residuals indicate mismatch between the lifted surrogate and the augmented state manifold. Spectral normalization bounds how lifted perturbations affect $\tilde{x}$, while training only on zero-padded data biases the inverse toward $\mathcal{S}$. These mechanisms keep r small in the operating region.

3.4 Loss Functions: A Simplified Landscape

A practical advantage of i-LiftProj is that architectural invertibility removes the need for several auxiliary losses used in LiftProj [6]. However, spectral normalization alone controls the conditioning of Φ and does not directly regularize the Jacobian of the lifted linearization residual. We therefore retain both data-driven and physics-informed linearization terms, together with a mild Jacobian regularizer for Φ. This is illustrated in Figure 2.

1) Proposed Objective (i-LiftProj). For compact notation, define the physical lifting induced by the augmented map as $\phi(x) := \Phi(x, \mathbf{0}_{N_L-n})$. For each data pair (x_t, u_t), with the next state generated as $x_{t+1} = f_{\mathrm{RK}}(x_t, u_t)$, define the lifted residual $r_{\mathrm{lin}}(x_t, u_t) := \phi\big(f_{\mathrm{RK}}(x_t, u_t)\big) - \big(A\phi(x_t) + Bu_t\big)$, where f_{RK} denotes the one-step Runge–Kutta discretization of the continuous-time dynamics. The training loss is:

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

where D_x and $D_{(x,u)}$ denote Jacobians with respect to the indicated variables, and $\|M\|_{1,1} := \sum_{i,j}|M_{ij}|$ denotes the entrywise ℓ_1 norm. The first term fits lifted one-step prediction, the second suppresses the sensitivity of the lifted residual, and the last promotes a near-isometric lifting on the sampled physical-state manifold.

2) Comparison with LiftProj. The original LiftProj [6] used linearization, autoencoding, and prediction losses in both data-driven and physics-informed forms:

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

In i-LiftProj, we retain only the two linearization terms, together with the conditioning regularizer in the proposed objective. The reconstruction losses (AE) are removed because inversion is computed by fixed-point iteration rather than a learned decoder, and the physical prediction losses (dyn) are omitted because the induced state-space error is governed by the conditioning of Φ and the lifted-model mismatch.

4 Convergence Analysis

The lifted linear model learned by i-LiftProj is only approximate, so the image of the true nonlinear dynamics manifold under the learned lifting need not coincide exactly with the linear subspace $\mathcal{H}$. Accordingly, the analysis is carried out with respect to a *surrogate dynamics manifold* induced by the learned lifting, and the discrepancy to the true manifold is tracked explicitly by a model-mismatch constant.

In this section only, we reinterpret y as the *augmented* trajectory variable $y = (\tilde{x}, u)$ so that the fixed-point inverse is analyzed before the final slicing step. We define the trajectory-space lifting map

$$\tilde{\Phi}(y) := (\Phi(\tilde{x}_0), \ldots, \Phi(\tilde{x}_N), u).$$

For brevity, we write $\kappa(\Phi) := L_{\tilde{\Phi}} L_{\tilde{\Phi}^{-1}}$ hereafter.

4.1 Surrogate Manifold and ADMM Interpretation

Define the surrogate dynamics manifold induced by the learned lifting as

$$\hat{\mathcal{C}}^{\mathrm{dyn}} := \big\{y \in \mathcal{D} \mid \tilde{\Phi}(y) \in \mathcal{H}\big\}, \tag{5}$$

where $\mathcal{D}$ is a compact neighborhood containing the iterates. The i-LiftProj retraction is then

$$\mathcal{T}_\Phi := \tilde{\Phi}^{-1} \circ \Pi_{\mathcal{H}} \circ \tilde{\Phi}. \tag{6}$$

For the local analysis below, we consider iterates for which the retracted points remain in $\mathcal{D}$; then $\mathcal{T}_\Phi(v) \in \hat{\mathcal{C}}^{\mathrm{dyn}}$ by construction.

To quantify the discrepancy between the true nonlinear dynamics manifold $\mathcal{C}^{\mathrm{dyn}}$ and its lifted linear surrogate, we define the local Koopman-model mismatch

$$\varepsilon_K := \sup_{y\in\mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D}} \operatorname{dist}\big(\tilde{\Phi}(y), \mathcal{H}\big). \tag{7}$$

The ADMM scheme analyzed below is the consensus problem associated with the surrogate manifold:

$$\operatorname{minimize}_{y,z}\ F(y) + I_{\hat{\mathcal{C}}^{\mathrm{dyn}}}(z), \quad \text{subject to } y = z. \tag{8}$$

Its augmented Lagrangian in scaled form is

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

The i-LiftProj algorithm iterates as follows:

1. $z^{(k+1)} \leftarrow \mathcal{T}_\Phi(y^{(k)} + \lambda^{(k)})$ [Lifted retraction]
2. $y^{(k+1)} \leftarrow \arg\min_y \hat{\mathcal{L}}_\rho(y, z^{(k+1)}, \lambda^{(k)})$ [Convex solve]
3. $\lambda^{(k+1)} \leftarrow \lambda^{(k)} + (y^{(k+1)} - z^{(k+1)})$ [Dual update]

4.2 Geometric Bounds on the Lifting Map

Assume that $\tilde{\Phi}$ is bi-Lipschitz on $\mathcal{D}$. Then we have that:

$$L_{\tilde{\Phi}^{-1}}^{-1}\|u - v\| \le \|\tilde{\Phi}(u) - \tilde{\Phi}(v)\| \le L_{\tilde{\Phi}}\|u - v\|, \quad \forall u, v \in \mathcal{D}.$$

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

$$\|\mathcal{T}_\Phi(v) - v\| \le \kappa(\Phi)\operatorname{dist}(v, \hat{\mathcal{C}}^{\mathrm{dyn}}), \tag{9}$$

$$\|\mathcal{T}_\Phi(v) - v\| \le \kappa(\Phi)\operatorname{dist}(v, \mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D}) + L_{\tilde{\Phi}^{-1}}\varepsilon_K. \tag{10}$$

Moreover,

$$\operatorname{dist}(v, \hat{\mathcal{C}}^{\mathrm{dyn}}) \le \operatorname{dist}(v, \mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D}) + L_{\tilde{\Phi}^{-1}}\varepsilon_K. \tag{11}$$

If, in addition, the relevant nearest point on $\mathcal{C}^{\mathrm{dyn}}$ lies in $\mathcal{D}$, then (10) *and* (11) *may be written with* $\operatorname{dist}(v, \mathcal{C}^{\mathrm{dyn}})$ *in place of* $\operatorname{dist}(v, \mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D})$.

Proof. By the Lipschitz continuity of $\tilde{\Phi}^{-1}$, we have:

$$\begin{aligned}\|\mathcal{T}_\Phi(v) - v\| &\le L_{\tilde{\Phi}^{-1}}\|\Pi_{\mathcal{H}}(\tilde{\Phi}(v)) - \tilde{\Phi}(v)\| \\ &= L_{\tilde{\Phi}^{-1}}\operatorname{dist}(\tilde{\Phi}(v), \mathcal{H}).\end{aligned} \tag{12}$$

Any $w \in \hat{\mathcal{C}}^{\mathrm{dyn}}$ satisfies $\tilde{\Phi}(w) \in \mathcal{H}$, by definition. Hence:

$$\operatorname{dist}(\tilde{\Phi}(v), \mathcal{H}) \le \|\tilde{\Phi}(v) - \tilde{\Phi}(w)\| \le L_{\tilde{\Phi}}\|v - w\|,$$

and taking the infimum over $w \in \hat{\mathcal{C}}^{\mathrm{dyn}}$ yields (9).

Next, using $\operatorname{dist}(a, S) \le \|a - b\| + \operatorname{dist}(b, S)$, with arbitrary $y \in \mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D}$ gives

$$\operatorname{dist}(\tilde{\Phi}(v),\mathcal{H}) \le \|\tilde{\Phi}(v) - \tilde{\Phi}(y)\| + \operatorname{dist}(\tilde{\Phi}(y),\mathcal{H}) \le L_{\tilde{\Phi}}\|v - y\| + \varepsilon_K.$$

Taking the infimum over $y \in \mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D}$ to obtain

$$\operatorname{dist}(\tilde{\Phi}(v), \mathcal{H}) \le L_{\tilde{\Phi}}\operatorname{dist}(v, \mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D}) + \varepsilon_K, \tag{13}$$

and combining (12) and (13) yields

$$\begin{aligned}\|\mathcal{T}_\Phi(v) - v\| &\le L_{\tilde{\Phi}^{-1}}\big(L_{\tilde{\Phi}}\operatorname{dist}(v, \mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D}) + \varepsilon_K\big) \\ &= \kappa(\Phi)\operatorname{dist}(v, \mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D}) + L_{\tilde{\Phi}^{-1}}\varepsilon_K,\end{aligned}$$

which proves (10).

Finally, let $y_* \in \arg\min_{y\in\mathcal{C}^{\mathrm{dyn}}\cap\mathcal{D}}\|v - y\|$ be a nearest point on the true dynamics manifold restricted to $\mathcal{D}$, and define $\hat{y}_* := \mathcal{T}_\Phi(y_*)$. Because $\hat{y}_* \in \hat{\mathcal{C}}^{\mathrm{dyn}}$, we have

$$\operatorname{dist}(v, \hat{\mathcal{C}}^{\mathrm{dyn}}) \le \|v - \hat{y}_*\|.$$

Also, since $y_* \in \mathcal{C}^{\text{dyn}} \cap \mathcal{D}$, we have $\text{dist}(y_*, \mathcal{C}^{\text{dyn}} \cap \mathcal{D}) = 0$. Applying (10) with $v = y_*$ therefore gives

$$\|\hat{y}_* - y_*\| = \|\mathcal{T}_\Phi(y_*) - y_*\| \le L_{\tilde{\Phi}^{-1}} \varepsilon_K.$$

Using the triangle inequality, we have that:

$$\begin{aligned}\text{dist}(v, \hat{\mathcal{C}}^{\text{dyn}}) &\le \|v - \hat{y}_*\| \le \|v - y_*\| + \|y_* - \hat{y}_*\| \\ &\le \text{dist}(v, \mathcal{C}^{\text{dyn}} \cap \mathcal{D}) + L_{\tilde{\Phi}^{-1}} \varepsilon_K,\end{aligned}$$

which proves (11). ∎

For the sliced implementation $\bar{\mathcal{T}}_\Phi := P_{\mathcal{S}} \circ \mathcal{T}_\Phi$, the bound in Lemma 1 becomes $\|\bar{\mathcal{T}}_\Phi(v) - v\| \le \kappa(\Phi)\,\text{dist}(v, \hat{\mathcal{C}}^{\text{dyn}}) + \varepsilon_r$ whenever $\|P_{\mathcal{S}} \mathcal{T}_\Phi(v) - \mathcal{T}_\Phi(v)\| \le \varepsilon_r$.

4.3 Residual-Floor Bound for i-LiftProj ADMM

The compact set $\mathcal{D}$ is a local operating neighborhood containing the warm-started ADMM iterates and the training-valid region of the learned lifting, and the theorem is conditional on the iterates remaining in this local region; state/control bounds, warm starts, or an explicit trust region can support this assumption in practice. The constant δ measures the maximum local distance of $v^{(k)}$ from the true dynamics manifold.

Theorem 2. *(Residual-Floor Bound under Surrogate Mismatch): Assume $F(y)$ is L_F-smooth and μ_F-strongly convex, and that the surrogate augmented Lagrangian $\hat{\mathcal{L}}_\rho(y,z,\lambda)$ is bounded from below. Let $w^{(k)}{=}(y^{(k)}, z^{(k)}, \lambda^{(k)})$ be generated by the i-LiftProj ADMM iteration, and define $v^{(k)} := y^{(k)} + \lambda^{(k)}$. Also assume that $v^{(k)} \in \mathcal{D}$, $\mathcal{T}_\Phi(v^{(k)}) \in \mathcal{D}$, and $\text{dist}(v^{(k)}, \mathcal{C}^{\text{dyn}} \cap \mathcal{D}) \le \delta$ for all k. Define*

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

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

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

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

Hence the i-LiftProj iterates admit an asymptotic residual floor controlled jointly by the conditioning of the learned lifting and the Koopman-model mismatch. In particular, the floor tightens as $\kappa(\Phi) \to 1$ and $\varepsilon_K \to 0$.

Proof. We first compare the augmented retraction step analyzed here, $z^{(k+1)} = \mathcal{T}_\Phi(v^{(k)})$, with the exact Euclidean projection onto the surrogate manifold, $\hat{z}_*^{(k+1)} = \Pi_{\hat{\mathcal{C}}^{\text{dyn}}}(v^{(k)})$. Because both points lie on the surrogate manifold, the difference of the augmented Lagrangian is only the quadratic penalty term:

$$\begin{aligned}\eta_k :=& \hat{\mathcal{L}}_\rho(y^{(k)}, z^{(k+1)}, \lambda^{(k)}) - \hat{\mathcal{L}}_\rho(y^{(k)}, \hat{z}_*^{(k+1)}, \lambda^{(k)}) \\ =& \tfrac{\rho}{2}\left(\|z^{(k+1)} - v^{(k)}\|^2 - \text{dist}(v^{(k)}, \hat{\mathcal{C}}^{\text{dyn}})^2\right).\end{aligned} \tag{16}$$

By Lemma 1, we have

$$\|z^{(k+1)} - v^{(k)}\| = \|\mathcal{T}_\Phi(v^{(k)}) - v^{(k)}\| \le \kappa(\Phi)\ \text{dist}(v^{(k)}, \hat{\mathcal{C}}^{\text{dyn}}),$$

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

Substituting these bounds into (16) gives

$$\eta_k \le \tfrac{\rho}{2}\left(\kappa(\Phi)^2 - 1\right) \text{dist}(v^{(k)}, \hat{\mathcal{C}}^{\text{dyn}})^2 \le \tfrac{\rho \hat{\delta}^2}{2}\left(\kappa(\Phi)^2 - 1\right) =: \bar{\eta}.$$

Thus the inexactness of the augmented retraction step is uniformly bounded by $\bar{\eta}$.

Next, because $\hat{z}_*^{(k+1)}$ is the exact minimizer of the surrogate z-subproblem, and because $z^{(k)} \in \hat{\mathcal{C}}^{\text{dyn}}$ from the previous iteration, we have

$$\begin{aligned}&\hat{\mathcal{L}}_\rho(y^{(k)}, z^{(k+1)}, \lambda^{(k)}) - \hat{\mathcal{L}}_\rho(y^{(k)}, z^{(k)}, \lambda^{(k)}) \\ &\le \hat{\mathcal{L}}_\rho(y^{(k)}, z^{(k+1)}, \lambda^{(k)}) - \hat{\mathcal{L}}_\rho(y^{(k)}, \hat{z}_*^{(k+1)}, \lambda^{(k)}) = \eta_k \le \bar{\eta}.\end{aligned} \tag{17}$$

For the y-step, note that

$$y^{(k+1)} = \arg\min_y \left\{F(y) + \tfrac{\rho}{2}\|y - z^{(k+1)} + \lambda^{(k)}\|^2\right\}.$$

Since F is μ_F-strongly convex, the function minimized above is $(\mu_F + \rho)$-strongly convex. Therefore the exact y-update yields the standard descent bound

$$\begin{aligned}&\hat{\mathcal{L}}_\rho(y^{(k+1)}, z^{(k+1)}, \lambda^{(k)}) - \hat{\mathcal{L}}_\rho(y^{(k)}, z^{(k+1)}, \lambda^{(k)}) \\ &\qquad \le -\tfrac{\mu_F + \rho}{2}\|y^{(k+1)} - y^{(k)}\|^2.\end{aligned} \tag{18}$$

For the dual update, using the optimality condition of the exact y-step, $\nabla F(y^{(k+1)}) + \rho\left(y^{(k+1)} - z^{(k+1)} + \lambda^{(k)}\right) = 0$, with $\lambda^{(k+1)} = \lambda^{(k)} + (y^{(k+1)} - z^{(k+1)})$ leads to $\rho\lambda^{(k+1)} = -\nabla F(y^{(k+1)})$, and the L_F-smoothness of $F(y)$ gives:

$$\begin{aligned}\|\lambda^{(k+1)} - \lambda^{(k)}\| &= \|-\left(\nabla F(y^{(k+1)}) - \nabla F(y^{(k)})\right)/\rho\| \\ &\le \tfrac{L_F}{\rho}\|y^{(k+1)} - y^{(k)}\|.\end{aligned}$$

Therefore the increase in the augmented Lagrangian in the dual update step is:

$$\begin{aligned}&\hat{\mathcal{L}}_\rho(y^{(k+1)}, z^{(k+1)}, \lambda^{(k+1)}) - \hat{\mathcal{L}}_\rho(y^{(k+1)}, z^{(k+1)}, \lambda^{(k)}) \\ &\qquad = \rho\|\lambda^{(k+1)} - \lambda^{(k)}\|^2 \le \tfrac{L_F^2}{\rho}\|y^{(k+1)} - y^{(k)}\|^2.\end{aligned} \tag{19}$$

Now summing (17), (18), and (19), we have:

$$\hat{\mathcal{L}}_\rho(w^{(k+1)}) - \hat{\mathcal{L}}_\rho(w^{(k)}) \le -\beta\|y^{(k+1)} - y^{(k)}\|^2 + \bar{\eta}. \tag{20}$$

Because $\hat{\mathcal{L}}_\rho$ is bounded below, we may sum (20) from $k = 0$ to K, divide by $K + 1$, and take lim inf to conclude:

$$\liminf_{k\to\infty} \|y^{(k+1)} - y^{(k)}\|^2 \le \tfrac{\bar{\eta}}{\beta} = \tfrac{\rho\hat{\delta}^2}{2\beta}\left(\kappa(\Phi)^2 - 1\right),$$

which proves (14). Finally, since $y^{(k+1)} - z^{(k+1)} = \lambda^{(k+1)} - \lambda^{(k)}$, the bound above on the dual increment gives:

$$\|y^{(k+1)} - z^{(k+1)}\|^2 \le \tfrac{L_F^2}{\rho^2}\|y^{(k+1)} - y^{(k)}\|^2,$$

and taking lim inf yields (15). ∎

This bound separates architecture-controlled conditioning from model mismatch: $\kappa(\Phi)^2 - 1$ is controlled by spectral normalization, whereas ε_K reflects lifted-model accuracy. After the spectral bound controls $\kappa(\Phi)^2 - 1$, the remaining problem-dependent scale is $\hat{\delta} = \delta + L_{\tilde{\Phi}^{-1}} \varepsilon_K$, while the sliced implementation further adds a measured post-processing perturbation. The dependence on ε_K motivates retaining the DD/PI linearization losses in Section 3.4.

Note that the theorem analyzes the ideal augmented retraction before the final slicing step and the sliced implementation can be treated as a bounded post-processing perturbation.

5 Numerical Experiments

We evaluate i-LiftProj on the 6-DoF rocket powered descent guidance (PDG) problem in [6]. We compare against LiftProj, which uses the same ADMM outer loop and lifted dimension, and against SCP as a reference solver.

Table 1: Performance on a 6-DoF PDG benchmark.

Model	Compute Time [sec]	Propagation Error	Fuel Consumption
SCP	2.4	5.49×10^{-3}	0.0495
LiftProj	1.5	5.26×10^{-3}	0.0492
i-LiftProj	2.2	4.97×10^{-4}	0.0495

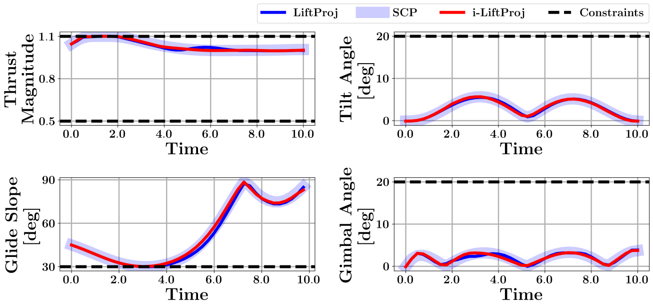


Figure 3: State/control trajectories and path-constraint satisfaction. i-LiftProj remains close to the SCP results.

Table 1 shows that i-LiftProj incurs an additional runtime cost due to fixed-point inversion, but achieves the lowest propagation error with comparable fuel consumption. Figure 3 shows constraint-satisfying trajectories that remain close to SCP, suggesting that the spectrally normalized lifting retains sufficient expressivity while improving trajectory regularity. Figure 4 shows faster initial and smoother overall primal-residual decay than LiftProj. These results support the conditioning-based interpretation of Section 4.

Here, we used $c = 0.5$ and $\varepsilon_{\text{fpi}} = 1.0\times10^{-7}$. The fixed-point inversion required 2.6 iterations per inverse call on average and at most 3 iterations over all ADMM iterations and time nodes. The trajectory-level augmentation residual, $e_{\text{slice}}^{(k)} = (\sum_t \|r_t^{(k)}\|^2 / \sum_t \|\tilde{x}_t^{(k)}\|^2)^{1/2}$, remained small, with $\max_k e_{\text{slice}}^{(k)} = 4.64\times10^{-5}$ and the ADMM-wise mean $\bar{e}_{\text{slice}} = 4.61\times10^{-5}$, as shown in Figure 5. These values are consistent with the absence of visible slicing-induced propagation error in Table 1 and with the lack of monotone growth in the primal residual curve in Figure 4. The runtime overhead is accompanied by improved numerical reliability and the conditioning bounds of Section 4.

In an additional randomized-initial-condition stress test over $N_{\text{MC}} = 100$ trials, LiftProj was sensitive to the removal of the SCP-calibrated feature box constraints used to keep the state/control feature values in a nominal data-supported range. In contrast, i-LiftProj returned converged feasible trajectories without such bounds, suggesting reduced sensitivity to off-nominal excursions.

6 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 autoencoder with an augmented i-ResNet and a decoder-free fixed-point inverse, the method provides a bi-Lipschitz ambient lifting map and removes learned-decoder reconstruction error. Analytically, the ADMM projection is interpreted through the surrogate manifold $\hat{\mathcal{C}}^{\text{dyn}} = \tilde{\Phi}^{-1}(\mathcal{H})$, yielding an asymptotic residual-floor bound controlled by the lifting condition number κ and the Koopman-model mismatch ε_K.

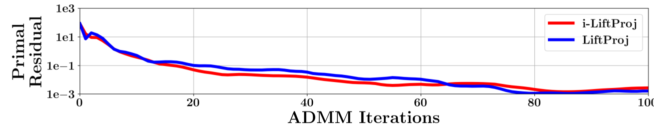


Figure 4: Primal-residual decay for LiftProj and i-LiftProj.

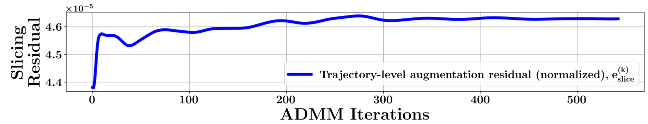


Figure 5: Trajectory-level augmentation residual.

Numerical experiments on 6-DoF PDG show that i-LiftProj preserves fuel use and constraint satisfaction while yielding lower propagation error, trajectories closer to SCP, and smoother early primal-residual decay than LiftProj. Although the compute time increases due to the fixed-point inverse, it remains comparable to SCP in this instance and is accompanied by improved numerical reliability and the conditioning bounds of the proposed lifting. Future work will extend the analysis to nonsmooth proximal constraints, and incorporate slicing residuals directly into the perturbation bound.

References

[1] L. Blackmore, “Autonomous precision landing of space rockets,” in *Frontiers of Eng.: Rep. on Leading-Edge Eng. from the 2016 Symp.*, vol. 46, pp. 15–20, The Bridge Washington, DC, USA, 2016.

[2] M. Szmuk, U. Eren, and B. Açıkmeşe, “Successive convexification for mars 6-dof powered descent landing guidance,” in *Proc. AIAA Guid. Navig. Control Conf.*, p. 1500, 2017.

[3] S. Boyd, N. Parikh, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[4] J. Choi, D. Lee, and J.-H. Kim, “Neural projection operators for real-time 6-dof powered descent guidance,” in *Proc. AIAA SCITECH 2026 Forum*, p. 1169, 2026.

[5] Y.-J. Kim, J. Choi, J. Choi, and J.-H. Kim, “A first-order approach for nonlinear optimal control under nonconvex constraints,” *Optim. Eng.*, vol. 27, 2026.

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

[7] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, “Regularisation of neural networks by enforcing Lipschitz continuity,” *Mach. Learn.*, vol. 110, pp. 393–416, 2021.

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

[9] M. Hong, Z.-Q. Luo, and M. Razaviyayn, “Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems,” *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, 2016.

[10] Y. Wang, W. Yin, and J. Zeng, “Global convergence of ADMM in nonconvex nonsmooth optimization,” *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, 2019.

[11] J. Behrmann, W. Grathwohl, *et al.*, “Invertible residual networks,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 573–582, PMLR, 2019.

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