Neural Lander: Stable Drone Landing Control using Learned Dynamics

Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, Soon-Jo Chung

Abstract

We present a novel deep-learning-based robust nonlinear controller (*Neural-Lander*) for stable quadrotor control during landing. Our approach blends together a nominal dynamics model coupled with a DNN that learns the high-order interactions, such as the complex interactions between the ground and multi-rotor airflow. Our approach has several attractive properties. First, by blending with a nominal dynamics model, our learning approach is very sample efficient, and our experiments show strong results using only 5 minutes of real-world training data. Second, by employing spectral normalization to constrain the DNN to have bounded Lipschitz behavior, we can design a nonlinear feedback linearization controller using the learned model and prove system stability with disturbance rejection. To the best of our knowledge, this is the first DNN-based nonlinear feedback controller with stability guarantees that can utilize arbitrarily large neural nets. Finally, this Lipschitz behavior also enables generalization so that the NN can make reliable predictions outside of its training distribution support. We demonstrate in live experiments that *Neural-Lander* can land smoothly in scenarios where the conventional approaches such as a PD controller cannot land precisely at all.

1 Introduction

Unmanned Aerial Vehicles (UAVs) require high precision control of aircraft positioning, especially during landing and take-off. This problem is challenging largely due to complex interactions of multi-rotor airflows with the ground (ground effect). Prior work in the aerial robotics community has largely focused on mathematical modeling (e.g. [1]). These ground-effect models are later used to approximate near-ground aerodynamics and integrated into controller design (e.g. [2]). However, existing theoretical ground effect models are derived based on steady-flow conditions, whereas most practical cases exhibit unsteady flow. Alternative approaches, such as integral or adaptive control methods, often suffer from slow response and delayed feedback. Given these limitations, the precision of existing fully automated systems for UAVs are still insufficient for landing and take-off, thereby necessitating the guidance of a human UAV operator during those phases.

To capture complex aerodynamic interactions without not being overly-constrained by conventional modeling assumptions, we take a learning approach to build a ground effect model using deep neural networks (NDNs). However, incorporating black-box models into a UAV controller faces two key challenges. First, due to high-dimensionality, NNs can be unstable and generate unpredictable output, which makes the system susceptible to instability in the feedback control loop. Second, DNNs are often difficult to analyze, which makes it difficult to design provably stable DNN-based controllers.

Contributions. We propose a learning-based controller, *Neural-Lander*, to improve precision of quadrotor landing with guaranteed stability. Our framework does not assume any fixed ground-effect model and instead learns it directly from real-world data collected from a quadrotor. Our overall dynamics model blends together a nominal dynamics model with a DNN to capture the high-order interaction effects between the ground and multi-rotor airflow.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

We train DNNs with spectral normalization (SN) of layer-wise weight matrices. We prove that SN of DNNs is stable for closed-loop control and that the proposed controller is globally exponentially stable under bounded learning errors. This is achieved by exploiting the Lipschitz bound of spectrally normalized DNNs. It has also been shown SN of DNNs leads to good generalization, i.e. stability in a learning-theoretic sense [3]. It is intriguing that SN simultaneously guarantees stability both in a learning-theoretic and a control-theoretic sense.

We empirically validate *Neural-Lander* for trajectory tracking of quadrotor during take-off, landing and near ground maneuvers. We show that *Neural-Lander* is able to land a quadrotor much more accurately than a naive PD controller with a pre-identified system. In particular, we show that *Neural-Lander* can decrease error in z direction from 0.13m to zero, and mitigate x and y drifts by 90% and 34% respectively, in 1D landing. Meanwhile, *Neural-Lander* can decrease z error from 0.12m to zero, in 3D landing. We also demonstrate that the learned ground-effect model can handle temporal dependency, and is an improvement over the steady-state theoretical models in use today. Our experiments also showcase the data efficiency of our approach, as the DNN component was trained on only 5 minutes of real-world training data, which highlights the potential of properly integrating nominal physics-based models with black-box learning.

2 Problem Statement: Precise Quadrotor Landing

Consider the following quadrotor dynamics. Given quadrotor states as global position $\mathbf{p} \in \mathbb{R}^3$, velocity $\mathbf{v} \in \mathbb{R}^3$, attitude rotation matrix $R \in \mathrm{SO}(3)$, and body angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v}, & m\dot{\mathbf{v}} &= m\mathbf{g} + R\mathbf{f}_u + \mathbf{f}_a \\ \dot{R} &= RS(\boldsymbol{\omega}), & J\dot{\boldsymbol{\omega}} &= J\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau}_u + \boldsymbol{\tau}_a \end{aligned} \tag{1}$$

where $\mathbf{g} = [0, 0, -g]^\top$ is the gravity vector; $\mathbf{f}_u = [0, 0, T]^\top$ and $\boldsymbol{\tau}_u = [\tau_x, \tau_y, \tau_z]^\top$ are the total thrust and body torques from four rotors predicted by a nominal model. Also, $\boldsymbol{\eta} = [T, \tau_x, \tau_y, \tau_z]^\top$ denotes the output wrench, and the linear equation $\mathbf{u} = [n_1^2, n_2^2, n_3^2, n_4^2]^\top$ relates the control input of squared motor speeds to the output wrench with its nominal relation given as $\boldsymbol{\eta} = B_0\mathbf{u}$:

$$B_0 = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & c_T l_{\text{arm}} & 0 & -c_T l_{\text{arm}} \\ -c_T l_{\text{arm}} & 0 & c_T l_{\text{arm}} & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \tag{2}$$

where c_T and c_Q denote some empirical coefficient values for force and torque generated by an individual rotor, and l_{arm} denotes the length of each rotor arm.

The key difficulty of precise landing is the influence of unknown disturbances $\mathbf{f}_a = [f_{a,x}, f_{a,y}, f_{a,z}]^\top$ and torques $\boldsymbol{\tau}_a = [\tau_{a,x}, \tau_{a,y}, \tau_{a,z}]^\top$, which originate from complex aerodynamic interactions between the quadrotor and the ground, such as the vertical aerodynamic force $f_{a,z}$. Also, as $\|\mathbf{v}\|$ increases, air drag will be exacerbated, which contributes to $\mathbf{f}_a$.

3 Neural Lander Controller Design and Nonlinear Stability Analysis

Learning High-Order Interactions. Our learning goal is to estimate $\hat{\mathbf{f}}_a(\boldsymbol{\zeta}, \mathbf{u})$ as the approximation to the disturbance aerodynamic forces, with $\boldsymbol{\zeta}$ being the partial states used as input features. In particular, we wish to estimate the residual forces not accounted for by a nominal dynamics model. The learning problem is thus a fairly standard supervised regression problem given real-world training data collected from a rotorcraft of interest flying close to the ground.

We employ spectral normalization to stablize DNN training by constraining the Lipschitz constant of the objective function. Spectral normalization has also been shown to generalize well [4] and in machine learning generalization is a notion of stability. Mathematically, the Lipschitz constant of a function $\|f\|_{\text{Lip}}$ is defined as the smallest value such that $\forall\, \mathbf{x}, \mathbf{x}' : \|f(\mathbf{x}) - f(\mathbf{x}')\|_2 / \|\mathbf{x} - \mathbf{x}'\|_2 \leq \|f\|_{\text{Lip}}$. It is known that the Lipschitz constant of a general differentiable function f is the maximum spectral norm (maximum singular value) of its gradient over its domain $\|f\|_{\text{Lip}} = \sup_{\mathbf{x}} \sigma(\nabla f(\mathbf{x}))$.

Learning-based Discrete-time Nonlinear Controller. Using our learned dynamics, the desired total force can be written as $\mathbf{f}_d = \bar{\mathbf{f}}_d - \hat{\mathbf{f}}_a(\boldsymbol{\zeta}, \mathbf{u})$, where $\bar{\mathbf{f}}_d$ is the desired force from the nominal PD

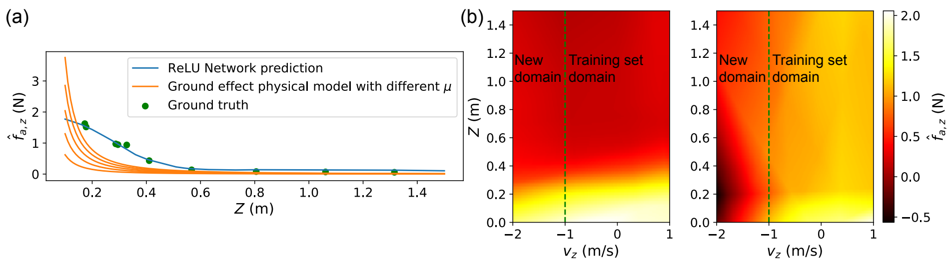

Figure 1: (a) Learned $\hat{f}_{a,z}$ compared to the ground effect model with respect to height z. $v_z = 0$ m/s and other dimensions of state are fixed. Ground truth points are from hovering data at different heights. (b) Heatmaps of learned $\hat{f}_{a,z}$ versus z and v_z, and other dimensions are fixed. (**Left**) Learned $\hat{f}_{a,z}$ from ReLU network with spectral normalization with $\|f\|_{\text{Lip}} = 1$. (**Right**) Learned $\hat{f}_{a,z}$ from ReLU network without spectral normalization with $\|f\|_{\text{Lip}} = 4.97$.

controller. Because of the dependency of $\hat{\mathbf{f}}_a$ on $\mathbf{u}$, the control synthesis problem here uses a non-affine control input for $\mathbf{u}$:

$$B_0 \mathbf{u} = \begin{bmatrix} \left(\bar{\mathbf{f}}_d - \hat{\mathbf{f}}_a(\boldsymbol{\zeta}, \mathbf{u})\right) \cdot \hat{k} \\ \boldsymbol{\tau}_d \end{bmatrix} \tag{3}$$

With $\boldsymbol{\eta}_d = [T_d, \boldsymbol{\tau}_d^\top]^\top$, We propose the following fixed-point iterative method for solving (3): $\mathbf{u}(t) = \mathbf{u}_k = B_0^{-1} \boldsymbol{\eta}_d(\mathbf{u}_{k-1})$, where $\mathbf{u}_{k-1}$ is the control input from the previous time-step.

Stability Guarantee. With a Lipschitz bound on $\hat{\mathbf{f}}_a(\boldsymbol{\zeta}, \mathbf{u})$, we can show convergence of our controller.

Lemma 3.1 *Define mapping* $\mathbf{u}_k = \mathcal{F}(\mathbf{u}_{k-1})$ *based on the above controller* $\mathbf{u}$*:*

$$\mathcal{F}(\mathbf{u}) = B_0^{-1} \begin{bmatrix} \left(\bar{\mathbf{f}}_d - \hat{\mathbf{f}}_a(\boldsymbol{\zeta}, \mathbf{u})\right) \cdot \hat{k} \\ \boldsymbol{\tau}_d \end{bmatrix}. \tag{4}$$

If $\hat{\mathbf{f}}_a(\boldsymbol{\zeta}, \mathbf{u})$ *is* L_a*-Lipschitz continuous, and* $\sigma(B_0^{-1}) \cdot L_a < 1$*; then* $\mathcal{F}(\cdot)$ *is a contraction mapping, and* $\mathbf{u}_k$ *converges to unique solution of* $\mathbf{u}^* = \mathcal{F}(\mathbf{u}^*)$.

Second, we can show stability rigorously under Assumptions 1-3.

Assumption 1 *The desired states along the position trajectory* $\mathbf{p}_d(t)$*,* $\dot{\mathbf{p}}_d(t)$*, and* $\ddot{\mathbf{p}}_d(t)$ *are bounded.*

Assumption 2 $\mathbf{u}$ *updates much faster than position controller.*

Note that the frequencies of attitude control ($> 100\,\text{Hz}$) and motor speed control ($> 5\,\text{kHz}$) are much higher than that of the position controller ($\approx 10\,\text{Hz}$) in practice.

Assumption 3 *The approximation error of* $\hat{\mathbf{f}}_a(\boldsymbol{\zeta}, \mathbf{u})$ *over the compact sets* $\mathcal{Z}, \mathcal{U}$ *is upper bounded by* $\epsilon_m = \sup_{\boldsymbol{\zeta} \in \mathcal{Z}, \mathbf{u} \in \mathcal{U}} \|\boldsymbol{\epsilon}(\boldsymbol{\zeta}, \mathbf{u})\|$*, where* $\boldsymbol{\epsilon}(\boldsymbol{\zeta}, \mathbf{u}) = \mathbf{f}_a(\boldsymbol{\zeta}, \mathbf{u}) - \hat{\mathbf{f}}_a(\boldsymbol{\zeta}, \mathbf{u})$.

4 Experiments

DNN Prediction Performance. Training data was collected from an Intel Aero quadrotor. In order to estimate the effect of disturbance force $\mathbf{f}_a$, we collected states and control inputs, while flying the drone close to the ground, manually controlled by an expert pilot, for 5 minutes. The inputs to our model are: $\hat{\mathbf{f}}_a = \hat{\mathbf{f}}_a(z, \mathbf{v}, R, \mathbf{u}) = \hat{\mathbf{f}}_a(\boldsymbol{\zeta}, \mathbf{u})$, where z, $\mathbf{v}$, R, $\mathbf{u}$ correspond to global height, global velocity, attitude, and control input. We estimate $\hat{\mathbf{f}}$ using a 4-layer ReLU network, with input and the output dimensions 12 and 3, respectively. We use spectral normalization (SN) so that the Lipschitz constant of our ReLU network is upper bounded. We also experimented with another ReLU network without spectral normalization, to further understand the benefits of SN.

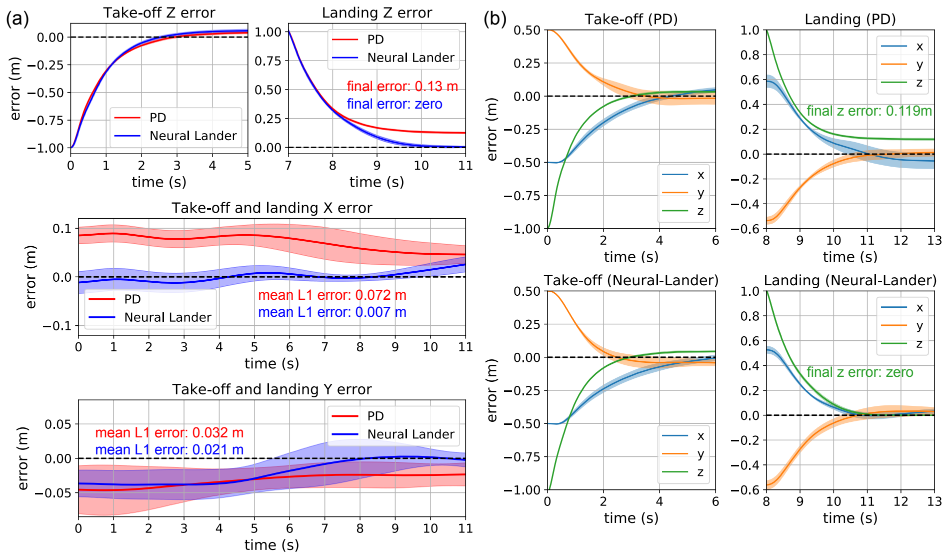

Figure 2: PD and *Neural-Lander* performance in (a) 1D and (b) 3D take-off and landing. Means (solid curves) and standard deviations (shaded areas) of 10 trajectories.

Fig. 1(a) shows the comparison between the estimated $\mathbf{f}_a$ from DNN and the theoretical ground effect model in [2, 5] as we vary the global height z (assuming $T = mg$ when $z = \infty$). We can see that our DNN can achieve much better estimates of the ground effects than the theoretical ground effect model. To understand the benefits of SN, we calculated learned $\hat{f}_{a,z}$ from the DNNs with and without SN. Fig. 1(b) shows the results. Note that -1 m/s to 1 m/s is covered in our training set but -2 m/s to -1 m/s is not. We see that: (1) **Ground effect**: $\hat{f}_{a,z}$ increases as z decreases, which is also shown in Fig. 1(a); (2) **Air drag**: $\hat{f}_{a,z}$ increases as the drone goes down ($v_z < 0$) and it decreases as the drone goes up ($v_z > 0$); (3) **Generalization**: the spectral normalized DNN is much smoother and can also generalize to new input domains not contained in the training set. In [4], the authors theoretically show that spectral normalization can provide tighter generalization guarantees on unseen data, which is consistent with our empirical results.

Control Performance. We used PD controller as the baseline controller and implemented both the baseline and *Neural-Lander*, as shown in Fig. 2. First we tested these two controller for the 1D take-off/landing task, i.e., moving the drone from $(0, 0, 0)$ to $(0, 0, 1)$ and then returning it to $(0, 0, 0)$. Second we compare the controllers for the 3D take-off/landing task, i.e., moving the drone from $(0, 0, 0)$ to $(0.5, -0.5, 1)$ and then returning it to $(0, 0, 0)$. We can conclude that the main benefits of our *Neural-Lander* are: (a) In both 1D and 3D cases, *Neural-Lander* can control the drone to precisely land on the ground surface while the baseline controller cannot land due to the ground effect. (b) In both 1D and 3D cases, *Neural-Lander* could mitigate drifts in x and y directions, as it also learned about non-dominant aerodynamics such as air drag. In experiments, we observed a naive un-normalized DNN ($\|f\|_{\mathrm{Lip}} = 247$) can even result in crash, which also implies the importance of spectral normalization. **Experimental video**: `https://youtu.be/C_K8MkC_SSQ`

References

[1] K. Nonaka and H. Sugizaki, "Integral sliding mode altitude control for a small model helicopter with ground effect compensation," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 202–207.

[2] L. Danjun, Z. Yan, S. Zongying, and L. Geng, "Autonomous landing of quadrotor based on ground effect modelling," in *Control Conference (CCC), 2015 34th Chinese*. IEEE, 2015, pp. 5647–5652.

[3] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[4] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 6240–6249.

[5] I. Cheeseman and W. Bennett, "The effect of ground on a helicopter rotor in forward flight," 1955.