Deploying Neural-Fly in the Field

Michael O'Connell[*,1], Guanya Shi[*,2], Xichen Shi[1], Kamyar Azizzadenesheli[3],
Animashree Anandkumar[1,3], Yisong Yue[1,4], and Soon-Jo Chung[1]

Abstract—**Neural-Fly is a learning-based control method that allows rapid online learning by incorporating a pre-trained representations of unmodeled aerodynamics with adaptive control. Not is Neural-Fly robust to different tasks, environments, and drones, but also it builds off a standard control architecture, creating a straightforward way to begin integrating learning-based control into safety critical applications. In this workshop, we will present the latest results for Neural-Fly on field robots, discuss implementation considerations, and show case other applications of Neural-Fly, such as fault tolerant control.**

I. Introduction

The proliferation of uninhabited aerial vehicles (UAVs) offers the prospect to revolutionize many areas. Applications range from drone delivery to drone rescue and search, and from urban air mobility to autonomous farming tools. These applications demand precise and agile control methods that can handle the complex aerodynamics while adapting to changing environmental and operating conditions. This creates a need for a control method that can rapidly adapt to new tasks, environments, and drones.

Our recent work, Neural-Fly [1], offers a solution, by pretraining a neural network to enable rapid and robust online learning of wind effects. This enables Neural-Fly to transfer between different tasks, environments, and drones while maintaining high performance Neural-Fly has been applied to deep-learning-based trajectory tracking control, and it has allowed quick adaptation to rapidly-changing wind conditions with centimeter-level position-error tracking of agile manuevers. Neural-Fly outperforms state-of-the-artcontrol methods when implemented in a readily-available UAV control architecture, PX4 [2]. Furthermore, Neural-Fly has demonstrated the ability to transfer control policies from one robot to another, and from limited range of constant wind speeds to a wide range of time-varying wind speeds. At this workshop, we will present the latest results for Neural-Fly deployment on field robots, some important implementation considerations, and show case other applications of Neural-Fly, such as fault tolerant control.

[*] Equal contribution to this work. [1] California Institute of Technology, [2] Robotics Institute, Carnegie Mellon University, [3] NVIDIA Corporation, [4] Latitude AI.
We thank J. Burdick and J.-J. E. Slotine for their helpful discussions. We thank M. Anderson for his help configuring the quadrotor platform, and M. Anderson and P. Spieler for their help troubleshooting hardware. We also thank N. Badillo and L. Pabon Madrid for help in experiments.
This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). This research was also conducted in part with funding from Raytheon Technologies. The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. The experiments reported in this article were conducted at Caltech's Center for Autonomous Systems and Technologies (CAST).

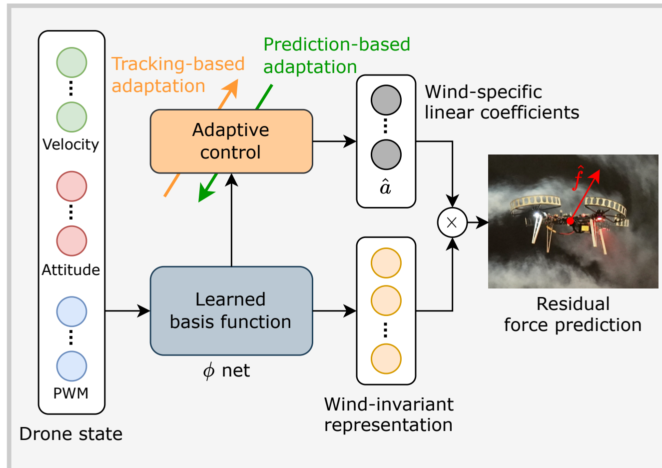

Fig. 1: **Neural-Fly design.** Neural-Fly learns a model of aerodynamics with linearly separated wind-variant and wind-invariant components. Since only part of the model must be updated in real time, Neural-Fly can quickly learn and adapt to new wind conditions.

II. Brief Overview of Neural-Fly

Our method has two main stages: an offline learning phase and an online adaptive control phase used as real-time online learning. For the offline learning phase, we have developed Domain Adversarially Invariant Meta-Learning (DAIML) that learns a wind-condition-independent deep neural network (DNN) representation of the aerodynamics in a data-efficient manner. The output of the DNN is treated as a set of basis functions that represent the aerodynamic effects. This representation is adapted to different wind conditions by updating a set of linear coefficients that mix the output of the DNN. DAIML is data efficient and uses only 12 total minutes of flight data in just 6 different wind conditions to train the DNN. DAIML uses spectral normalization [3], [4] to control the Lipschitz property of the DNN to improve generalization to unseen data and provide closed-loop stability and robustness guarantees. As seen in Fig. 3, training data generated in different wind conditions can have high correlation between the actual trajectory of the vehicle and the wind condition present. DAIML uses a discriminative network to ensure that the learned representation is wind-invariant and to prevent overfitting to the highly correlated training data. The result is that DAIML trains a concise representation of the aerodynamics that is both data efficient and generalizes well to new wind conditions.

Fig. 2: **Agile flight through narrow gates.** Neural-Fly was tested in the Caltech Real Weather Wind Tunnel. These panels show the moment the UAV passed through a narrow gate, only slightly wider than the UAV itself.

For the online adaptive control phase, Neural-Fly uses a robust and fast adaptive control law to update the model for new wind conditions. The adaptation algorithm is built from a Kalman Filter [5], [6] estimator of the linear coefficients, $a(w)$, which provides robustness and regularization properties. The Kalman Filter is augmented with a tracking error term to make the closed loop dynamics stable during rapid adaptation. The combination of the prediction error based Kalman filter and tracking error based adaptation term makes this approach a composite adaptive control law, and effectively guarantees fast and stable adaptation to any wind condition and robustness against imperfect learning. The speed of adaptation is further aided by the concise representation learned from DAIML.

III. Experimental Validation

Neural-Fly was tested on an agile figure-8 trajectory and compared with several methods that represent the state of art in quadrotor control. Each method was tested in a variety of wind conditions, including wind speeds inside the range of wind speeds seen in training (0 m/s to 4.2 m/s), and wind speeds outside the range of wind speeds seen in training (8.5 m/s to 12.1 m/s), and time varying wind speeds that break the constant wind-speed assumption made during training ($8.5 + 2.4\sin(t)\,\mathrm{m/s}$).

When measuring position tracking errors, we observe that our Neural-Fly method outperforms state-of-the-art flight controllers in all wind conditions. Using Neural-Fly, we report an average improvement of 66 % over a nonlinear tracking controller [7], 42 % over an $\mathcal{L}_1$ adaptive controller [8], [9], and 35 % over an Incremental Nonlinear Dynamics Inversion (INDI) controller [10].

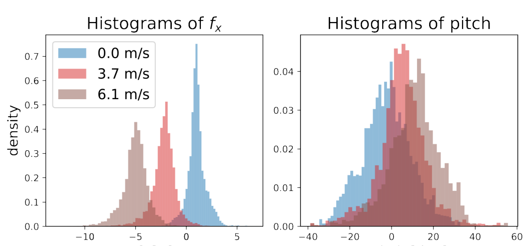

Fig. 3: **Input-output correlation in the training data.** Histograms showing data distributions in different wind conditions, showing that the shift in wind conditions causes a distribution shift in the input **Left:** distributions of the x-component of the wind-effect force, f_x. **Right:** distributions of the pitch, a component of the state used as an input to the learning model.

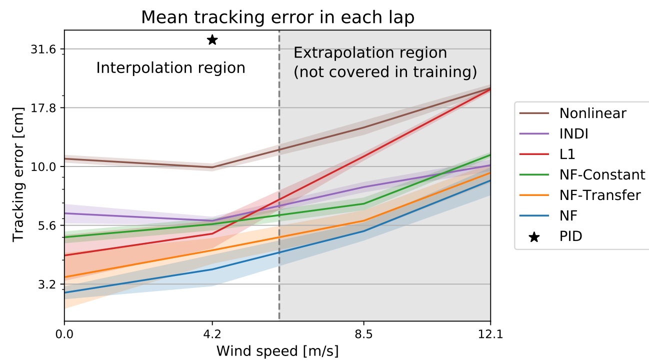

Fig. 4: **Mean tracking errors in different wind conditions.** Solid lines show the mean error over 6 laps and the shade areas show standard deviation of the mean error on each lap.

Neural-Fly can generalize to new conditions, as demonstrated by its performance in wind speeds outside the training range and in time varying wind speeds. Neural-Fly-Transfer was trained on data collected using a different quadrotor without motor tilt, different size, and a different propeller configuration, but Neural-Fly-Transfer maintains nearly as good of performance as Neural-Fly. Thus, Neural-Fly-Transfer demonstrates that Neural-Fly is robust to changes in vehicle configuration and modeling errors.

IV. Conclusion

Neural-Fly is formulated generally for all robotic systems described by the Euler-Langrange equation, and should be applicable to a wide range of robotic systems. Neural-Fly demonstrates a new paradigm for designing adaptable controllers that can be trained once and then used to control a wide range of vehicles.

References

[1] M. O'Connell, G. Shi, X. Shi, *et al.*, "Neural-Fly enables rapid learning for agile flight in strong winds," *Sci-*

ence Robotics, May 4, 2022. DOI: 10.1126/scirobotics.abm6597. (visited on 05/13/2022).

[2] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, “PIXHAWK -A micro aerial vehicle design for autonomous flight using onboard computer vision,” *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, Aug. 2012, ISSN: 0929-5593. DOI: 10.1007/s10514-012-9281-4. (visited on 08/31/2021).

[3] G. Shi, X. Shi, M. O’Connell, *et al.*, “Neural Lander: Stable Drone Landing Control using Learned Dynamics,” *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9784–9790, May 2019. DOI: 10.1109/ICRA.2019.8794351. arXiv: 1811.08027. (visited on 09/02/2021).

[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*, vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/b22b257ad0519d4500539da3c8bcf4dd-Abstract.html (visited on 11/18/2022).

[5] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1, 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552. (visited on 09/21/2021).

[6] R. E. Kalman and R. S. Bucy, “New Results in Linear Filtering and Prediction Theory,” *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, Mar. 1, 1961, ISSN: 0021-9223. DOI: 10.1115/1.3658902. (visited on 09/21/2021).

[7] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 2520–2525. DOI: 10.1109/ICRA.2011.5980409.

[8] S. Mallikarjunan, B. Nesbitt, E. Kharisov, E. Xargay, N. Hovakimyan, and C. Cao, “L1 Adaptive Controller for Attitude Control of Multirotors,” in *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, Minnesota: American Institute of Aeronautics and Astronautics, Aug. 13, 2012, ISBN: 978-1-60086-938-9. DOI: 10.2514/6.2012-4831. (visited on 03/04/2022).

[9] D. Hanover, P. Foehn, S. Sun, E. Kaufmann, and D. Scaramuzza, “Performance, Precision, and Payloads: Adaptive Nonlinear MPC for Quadrotors,” Sep. 9, 2021. arXiv: 2109.04210 [cs]. [Online]. Available: http://arxiv.org/abs/2109.04210 (visited on 09/16/2021).

[10] E. Tal and S. Karaman, “Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1203–1218, May 2021, ISSN: 1558-0865. DOI: 10.1109/TCST.2020.3001117.