

Learning-based Air Data System for Safe and Efficient Control of Fixed-wing Aerial Vehicles

Krzysztof Choromanski¹, Vikas Sindhwani², Brandon Jones³, Damien Jourdan⁴, Maciej Chociej⁵, Byron Boots⁶

Abstract—We develop an air data system for aerial robots executing high-speed outdoor missions subject to significant aerodynamic forces on their bodies. The system is based on a combination of Extended Kalman Filtering (EKF) and autoregressive feedforward Neural Networks, relying only on IMU sensors and GPS. This eliminates the need to instrument the vehicle with Pitot tubes and mechanical vanes, reducing associated cost, weight, maintenance requirements and likelihood of catastrophic mechanical failures. The system is trained to clone the behaviour of Pitot-tube measurements on thousands of instrumented simulated and real flights, and does not require a vehicle aerodynamics model. We demonstrate that safe guidance and navigation is possible in executing complex maneuvers in the presence of wind gusts without relying on airspeed sensors. We also demonstrate accuracy enhancements from successful “simulation-to-reality” transfer and dataset aggregation techniques to correct for training-test distribution mismatches when the air-data system and the control stack operate in closed loop.

I. INTRODUCTION

Aerial robots [1], [2] equipped with powerful avionics, on-board sensors, GPS modules and cameras have the potential to accomplish a variety of autonomous navigation and perceptual reasoning tasks from a bird’s-eye view. Compelling applications include emergency response [3], search-and-rescue, goods delivery, transportation, precision agriculture [4], industrial inspection, environment preservation and 3D photography. In a recent study [3], GPS-equipped 8-rotor aerial vehicles flying autonomously over a few miles, were able to deliver equipment to simulated out-of-hospital cardiac arrest sites almost four times faster than conventional emergency medical services. Such applications, combined with the affordability of smartphone-quality hardware components, has led to a surge of interest in low-cost unmanned aerial vehicles (UAVs).

Typical multi-rotor UAVs (e.g. quadcopters) generally do not require knowledge of airspeed to operate safely, since they use their motors to generate the required forces and

moments. In this case, aerodynamic forces caused by non-zero airspeeds are usually small enough to be treated as disturbances by the control system. However this is not the case for other aircraft designs, from both a safety and performance standpoint.

Fixed-wing or hybrid aircraft rely on structural features (wings, tails, control surfaces) to generate aerodynamic forces and moments, quantities that will vary linearly or quadratically with airspeed. Errors in the airspeed estimate will cause the flight control system to under- or over-predict forces and moments. For example, over-estimating airspeed will lead to an over-prediction of the effectiveness of control surfaces such as elevators, and could also lead to stalling the main lifting surfaces (wings). In both cases loss of control can ensue. Aircraft are also designed to operate most efficiently at a best cruise airspeed, which will maximize the distance that the aircraft can fly. Errors in the airspeed estimate will cause the aircraft to cruise at suboptimal airspeed, causing a reduction in range and possibly preventing it from completing its mission.

It is also worth noting that GPS velocity is not a good replacement for airspeed, since winds will cause GPS velocity and airspeed to differ, sometimes very significantly. Again, this is most critical for fixed-wing aircraft.

The accurate sensing of relative motion with respect to air is therefore critical for safe and efficient control of fixed-wing aerial vehicles executing outdoor missions in variable wind conditions. An on-board air data system outputs an estimate of the speed and orientation of the vehicle relative to the air mass in which it is flying. These quantities are the airspeed (V_T), angle of attack α , and the sideslip angle β (described precisely later). Passenger aircraft measure this triplet using multiple redundant Pitot tubes and mechanical vanes fitted near the nose of the aircraft. On small UAVs, however, such sensors disproportionately increase cost, weight and maintenance requirements. Air data systems relying on these sensors require additional design, mechanical integration, and calibration to be accurate [5]. Once operational, such instrumentation is susceptible to moisture and blockage, requiring continual maintenance to ensure reliability. Failure of the air data system can have catastrophic consequences [6], such as the 2009 crash of Air France Flight 447 into the Atlantic ocean caused by pilot disorientation due to erratic airspeed readings from iced Pitot tubes.

Small, inexpensive airspeed sensors can be purchased off-the-shelf for small UAVs. However these sensors are not

Krzysztof Choromanski is a research scientist at Google Brain Team, New York, NY 10011, USA, kchoro@google.com

Vikas Sindhwani is a research scientist at Google Brain Team, New York, NY 10011, USA, sindhwani@google.com

Brandon Jones is a researcher at X, USA, brnjones@x.team

Damien Jourdan is a researcher at X, USA, jourdan@x.team

Maciej Chociej was at Google Brain Team, New York, NY 10011, USA, maciekc@google.com

Byron Boots was visiting faculty in Google Brain Team, New York. He is assistant professor at Georgia Tech, Atlanta, GA 30332-0280. boots@cc.gatech.edu

designed to operate in rainy, icing, or dusty conditions, and will fail unexpectedly when exposed to such conditions. Due to the small size of the sensor, these challenges are not as easily solved as for passenger aircraft, which can afford costly heated tubes with larger diameters. Designing a small, inexpensive Pitot sensor that can withstand all operational environments is very challenging.

Extended Kalman Filtering (EKF) based traditional air data systems suffer from needing to adequately model system dynamics. Building an accurate model of the system is challenging, especially for small UAVs where the aerodynamic interactions are more complex than on larger fixed-wing aircraft (e.g. rotor cross-flow, flow around small structures). This requires pushing the boundary of computational fluid dynamics tools, or performing complex and expensive measurement campaigns using wind tunnels. In contrast, collecting flight data is easily achievable with small autonomous systems (as opposed to large aircraft), and something most small UAV developers are able to do. It is therefore desirable to use raw flight data to produce a direct airspeed measurement, rather than taking the intermediate step of modeling the underlying dynamics of the system.

Motivated by the observations above, in this paper, we develop an alternative air data system based on machine learning. We demonstrate that a neural network trained to clone the behavior of the Pitot tube on thousands of simulated and real flights can learn to sense relative air motion reliably enough to enable safe guidance, navigation and control of aerial vehicles. The neural network learns to map the evolving history of inertial motion of the vehicle in response to control commands, to an estimate of the current airspeed. It relies on measurements from only the IMU (Inertial Measurement Unit) sensors and GPS (Global Positioning System).

The airspeed prediction is converted to dynamic pressure and used by on-board controllers to navigate the vehicle along the trajectory of a payload delivery mission. All our experiments are conducted on an unconventional Vertical Takeoff and Landing (VTOL) aircraft with lift motors and fixed-wing forward flight capability. We consider a variety of real and simulated missions where the vehicle must fly through specific waypoints starting from a base location, dropping off a payload at a designated destination and returning to the base. We find that the airspeed predictions are accurate to within 2 meters per second for nearly the entirety of the flight time. We test the robustness of proposed air-data system in settings that exercise the controller’s use of air-speed, e.g. in missions requiring several turns in the presence of wind gusts, and test missions that are explicitly manually piloted to create unfamiliar mission profiles. Our approach is completely model-free and is expected to generalize to other types of UAVs and outdoor mission specifications. Additionally, we demonstrate two promising mechanisms to boost the accuracy of the system: (i) successful “simulation-to-reality” transfer where large amounts of training data under varying wind conditions can be used from flight simulators, and (ii) a dataset aggregation mechanism [7], [8]

that boosts baseline models by correcting for training-test mismatch when the air-data system and the control stack operate in closed loop.

The goal of our exposition is to provide an accessible overview to both aerospace engineering and machine learning communities. We provide a brisk overview of related work followed by a description of our approach and extensive empirical analysis (further detailed in supplementary material).

II. RELATED WORK

Machine learning has proven to be an effective tool for modeling the physical properties and dynamics of ground and air vehicles directly from data. Abbeel et al. [9] and Ross and Bagnell [10] learned models of helicopter dynamics directly from trajectories and then used the learned models for helicopter control. More recently, Williams et al. [11] learned neural network models of vehicle dynamics for model predictive control of a fast-moving autonomous ground vehicle. Much recent work in machine learning has focused on the problem of learning for sequential prediction and decision making: DAGger [7], Data as Demonstrator [8], and similar algorithms [12] have been used to directly learn control policies for aerial vehicles from human demonstrations [13], as well as train functions for filtering and state estimation [14]. Venkatraman et al. [15] and Sun et al. [16] considered the problem of learning to predict unknown sensor values from partial state information. They showed that dataset aggregation [7] could be used for quadrotor attitude estimation [15] when the attitude was available during training but not testing and that the learned model systematically outperformed a hand-tuned complementary filter [17]. Finally, machine learning has also been used to learn models in simulation and then transfer those models to real world systems. While this can be difficult due to the so-called “Reality Gap” [18], the fact that simulators rarely capture the full complexity of the real world, recent work on policy learning for unmanned aerial vehicles has demonstrated that crossing this gap is sometimes possible [19].

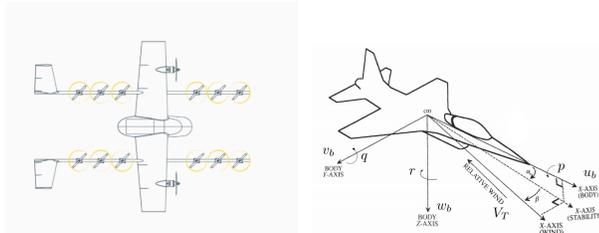
In aerospace engineering, a significant amount of previous work has focused on estimating angle of attack $\hat{\alpha}$ and sideslip angle $\hat{\beta}$ using a direct measurement of the true velocity V_T [20], [21], [22]. A synthetic air-data system was reported to be used for closed-loop control on the X-45A Joint Unmanned Combat Air system program [23]. This method used a dynamic pressure measurement and the full 6-DOF non-linear aerodynamic model of an aircraft in an EKF framework to estimate α and β only. A few methods provide $\hat{\alpha}$, $\hat{\beta}$, and \hat{V}_T without airspeed measurement. These techniques typically rely upon an inertial measurement of velocity, attitudes, and an aircraft dynamics model. A backup synthetic air data system was proposed for the X-38 Crew return vehicle using inertial measurements and an aerodynamic model to estimate α , β , and V_T [24]. A severe limitation of this approach was the assumption of a reference trajectory: only air data perturbations from the reference were computed limiting filtering performance beyond trim.

Lie et al. [25] proposed a cascaded Extended Kalman Filter (EKF) architecture to prevent aircraft dynamic modeling errors from corrupting normal states. A lower level EKF was used to estimate position, ground speed, attitude, and sensor biases and a higher level EKF was used, along with aircraft dynamic models, to produce $\hat{\alpha}$, $\hat{\beta}$, and \hat{V}_T . The effects of dynamic model uncertainty were shown to be mitigated by capturing specific non-linear aspects in the linear model, improving the performance of the estimator during deviations from the trim condition [26]. The filter demonstrated good performance at off-trim attitudes, but not off-trim velocities. Finally, Shaqura and Claudel [27] used a “hybrid” approach to estimate α , β , and V_a . The non-linear aircraft model was decomposed into a finite number of linear modes. The mode that matched the flight condition was selected at each time-step and linear regression was performed on this mode to estimate the air data parameters. This hybrid method demonstrated a computational performance improvement of 2x over an equivalent EKF implementation.

III. LEARNING TO SENSE THE AIR

A. Aircraft Description, Navigation and Control

In this paper, a prototype fixed-wing Vertical Takeoff and Landing (VTOL) aircraft designed for a package delivery mission is used to demonstrate the airspeed estimation technique. A schematic of the vehicle is shown on the left in the figure below. An array of 12 vertically mounted electric motors provide thrust for hovering flight. Two forward thrust motors, two ailerons, and two ruddervators are used primarily for cruise flight. This hybrid configuration provides a flight envelope across a full range of airspeeds from 0 m/s to best aerodynamic cruise.



Aircraft performance, stability, and control is highly dependent on in-flight aerodynamic forces and moments imparted on the air vehicle. These aerodynamic forces, $F_{\text{aero}} \in \mathbf{R}^{1 \times 3}$, can be expressed as $F_{\text{aero}} = f(Q, \alpha, \beta, \omega, u)$, with dynamic pressure, $Q = \frac{1}{2} \rho V_T^2$ where ρ denotes air-density, true airspeed, $V_T \in \mathbf{R}$, angle of attack $\alpha \in \mathbf{R}$, sideslip $\beta \in \mathbf{R}$, angular rates about a body-fixed frame of reference $\omega_b = [p \ q \ r]^T$, and actuator inputs $u \in \mathbf{R}^n$, for n actuators [28]. True airspeed, V_T , is the velocity magnitude of the relative air mass, while α and β are the relative angles of the wind with respect to the aircraft body-axis. An overview of air data parameters α , β , and V_T in the context of body, stability, and wind axes is presented in the schematic on the right in the figure above, adapted from [28]. Additionally, the North-East-Down (NED) reference frame

refers to a coordinate system whose origin is typically the missions home location with axes aligned with the northern and eastern axis, with the vertical axis pointing downwards towards the center of the earth.

An air-data system is designed to provide real-time observations of V_T , α , and β . These observations are fundamental to aircraft operation, as they are often used in automatic control laws, maintaining a vehicle state within a safe range, and targeting conditions to maximize flight efficiency. A typical air data system directly measures Q , α , and β with sensors such as a Pitot tubes [29], flush mounted pressure transducers, and wind vanes.

1) *Package Delivery Missions*: Each mission is defined by a TAKEOFF action at the home location followed by a combination of CRUISE, HOVER, PICKUP, DELIVER and LAND actions at designated waypoints specified by latitude-longitude pairs. The missions are executed in real flights as well as in flight simulators. In the latter case, a model of a Pitot tube incorporating errors such as random noise and bias is used for airspeed measurements. The simulator allows the performance of the aircraft to be studied under variable wind conditions and levels of sensor noise.

2) *Vehicle State Estimation*: A flight control system requires information about the current state of the aircraft to track mission commands (e.g. following a path at a given best-cruise airspeed). The state of interest is defined as:

- 3D Position (e.g. in the North/East/Down (NED) frame, with origin at the take-off location)
- NED inertial velocity
- Rotation from body to NED frame (e.g. Euler angles: roll ϕ , pitch θ , and true heading ψ)
- body rotation rates (p , q , and r)
- body accelerations (a_x , a_y , and a_z)
- True airspeed V_T

The state is estimated by blending the measurements from a set of sensors. For a small fixed-wing UAV, the minimum set of sensors typically includes:

- Inertial Measurement Unit (IMU, consisting of 3 accelerometers and 3 rate gyros)
- GPS receiver
- 3D magnetometer
- Static pressure sensor
- Pitot tube with dynamic pressure sensor

This information is blended through a state estimator such as an Extended Kalman Filter (EKF), typically running between 100 Hz and 400Hz. In many cases true airspeed itself is derived directly from the Pitot tube measurement, and does not use information from other sensors. It is computed according to, $V_T = \sqrt{2Q/\rho}$, where Q is the measured differential pressure in Pascals and ρ is air density in kg/m^3 . In the absence of an outside air temperature sensor, air density is looked-up based on altitude, assuming standard atmospheric conditions [29].

3) *Control*: The position, velocity, and attitude estimates from the EKF are compared with commands generated by a high-level mission planning system. The control algorithm generates actuator commands to reduce errors between the state estimate and commands. The controller incorporates a real-time airspeed estimate to properly allocate control between individual hover motors and aerodynamic control

surfaces throughout the airspeed envelope. For example, ailerons are not effective in controlling the roll axis at low airspeed. Therefore, the control algorithm will allocate roll control to hover motors rather than ailerons based on the airspeed estimate.

B. Learning-based Air Data System

1) *Cascaded Architecture*: The proposed air data system is a cascade of an EKF state estimator followed by a neural network autoregressive predictor, as illustrated in Figure 1. The first stage EKF outputs the velocity and acceleration of the vehicle relative to the ground, and its orientation and rate of rotation, based on sensor measurements from the GPS and IMU. In this architecture, the air data system does not rely on the knowledge of the aircraft aerodynamic model, and the cascaded design implies that airspeed estimation errors cannot corrupt fundamental safety critical state estimation of attitudes, velocity, and position.

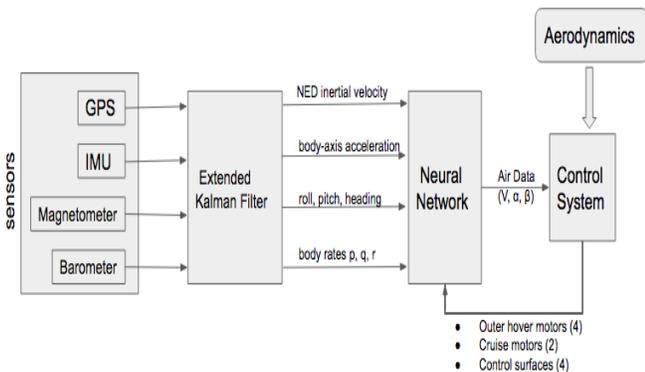


Fig. 1. Hybrid EKF-Neural Network Air Data System

The neural network airspeed predictor and the control system form a feedback loop. A 22-dimensional feature vector is formed using the following components: vehicle state as summarized by (i) Attitude ϕ (roll), θ (pitch), and ψ (true heading), (ii) Body rates p, q, r , (iii) Body acceleration, (iv) NED inertial velocity as measured by the GPS receiver, together with actuator commands issued by the control system which include (i) 4 Hover motor commands: leftfront, rightfront, leftrear, rightrear, (ii) 2 Cruise motor commands: cruiseleft, cruiseright and (iii) 4 Control surface commands: leftaileron, rightaileron, leftruddervator, rightruddervator.

These feature vectors are accumulated over a sliding window of fixed length $\text{windowlength} = 50$ samples. Each mission typically generates 20-50 thousand sliding window chunks. The sampling rate is 0.01 second so that the sliding window lengths are 0.5 seconds long. Each chunk is flattened into a $\text{windowlength} \times 22 = 1100$ -dimensional vector by concatenating all 22-dimensional feature vectors. As flight data streams in, a circular buffer maintains vehicle state and actuator commands over the sliding window. The resulting input vector is fed to the neural network to generate an airspeed prediction, which is then consumed by the control stack, completing the loop.

2) *Cloning the Behavior of the Pitot Tube*: The neural network is trained to mimic the Pitot tube on data collected from instrumented real and simulated flights. In this paper, we demonstrate the system on airspeed prediction though angle of attack and sideslip angle can also be similarly handled. The training data is of the form $\{(x_t^i, V_t^i), t = 1 \dots T^i, i = 1 \dots N\}$, where i indexes training missions, x_t denotes vehicle states and actuator inputs accumulated over a sliding window ending at time t , and V_t denotes the airspeed measurement based on the Pitot tube. The neural network parameters θ are obtained by minimizing the squared loss,

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N \sum_{t=1}^{T^i} (V_t^i - f_{\theta}(x_t^i))^2 \quad (1)$$

The learning process is regularized with dropout training.

The neural network airspeed predictor is a fully connected architecture with one hidden layer of size $h = 1000$ and ReLU activation functions. We experimented with deeper networks as well, but did not find appreciable gains.

Training Infrastructure: The loss is minimized using stochastic gradient descent using the ADAM optimizer [30] in TensorFlow [31] with minibatch size of 100. The network weights are initialized from a truncated Normal distribution with standard deviation 0.1. Large-scale training is distributed over a cluster of 100 machines, (Tesla k20 GPU accelerators, 80G RAM). A continuous training infrastructure is set up to easily retrain on incoming instrumented flight logs.

IV. EMPIRICAL ANALYSIS

A. Airspeed Prediction: Standalone Accuracy

In this section we study how well our machine learning-based air data system mimics Pitot tube airspeeds on a collection of real and simulated test missions. These results are “standalone” in the sense that the control system still relies on the Pitot tube. We also report the effectiveness of “simulation-to-reality” transfer in this setting. In section IV-B, we disable the Pitot tube and study end-to-end mission performance where the controller uses air data estimates from our model. Please note that an exhaustive set of experimental results are included in our supplementary material.

Metrics: In the experiments below we use the following metrics: average error ϵ_{ave} , mean squared error ϵ_{mse} , maximum error L_{∞} , percentage of time with error less than 1.5 m/s as well as the so-called cdf curves. The cdf curve at point (x, y) means that for the y -fraction of flight time the airspeed prediction error is at most x . Neural network models were trained on data with labels given as Pitot tube measurements or groundtruth airspeeds (the latter one for certain simulated training missions). If not mentioned otherwise, we assume that labels are Pitot tube measurements.

1) *Different train-test modes*: Sim2Sim, Real2Real, Sim2Real and Hybrid2Real: Figures 2 and 3 present results for four combinations of train-test scenarios depending on whether the training and test data is drawn from simulated

or real missions. In general, errors ϵ_{ave} , ϵ_{mse} and L_∞ were computed for different test missions and averaged over all of them. For the Sim2Sim setting the averaged test errors are: $\epsilon_{ave} = 1.74$ m/s, $\epsilon_{mse} = 4.79$ m²/s² and $L_\infty = 7.10$ m/s. For the Real2Real setting the averaged test errors are: $\epsilon_{ave} = 2.25$ m/s, $\epsilon_{mse} = 16.37$ m²/s² and $L_\infty = 7.23$ m/s. For the Sim2Real setting the averaged test errors are: $\epsilon_{ave} = 3.15$ m/s, $\epsilon_{mse} = 15.13$ m²/s² and $L_\infty = 8.2$ m/s. For the Hybrid2Real setting the averaged test errors are: $\epsilon_{ave} = 2.17$ m/s, $\epsilon_{mse} = 8.46$ m²/s² and $L_\infty = 8.5$ m/s.

High-quality airspeed predictions are obtained throughout the flight duration (see Figure 2). Training on simulated data helps improve performance on real flights.

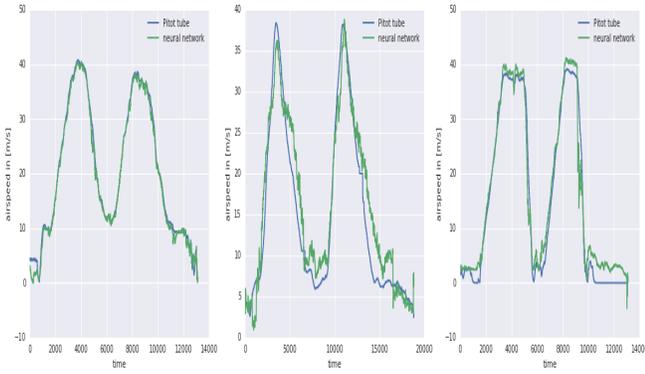


Fig. 2. Comparison of airspeed predicted by the neural network with Pitot tube measurements for different train-test scenarios: Real2Real (left), Sim2Real (middle) and Hybrid2Real (right).

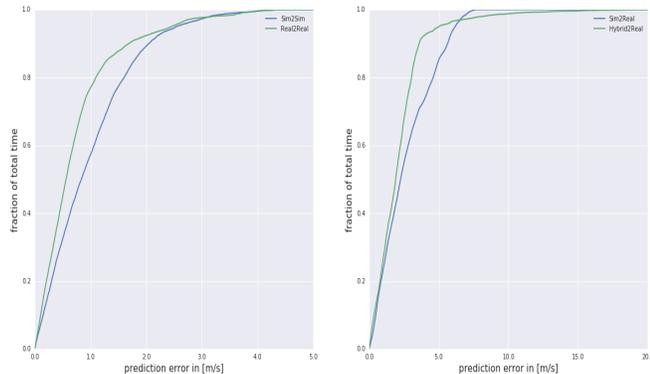


Fig. 3. Cdf curves for all four train-test scenarios: Sim2Sim, Real2Real, Sim2Real and Hybrid2Real. Left: training and testing in the same environment (Sim2Sim, Real2Real). Right: sim data participates in training while testing is conducted on real data.

In the above experiments, the 5-epoch neural network training was conducted on datasets consisting of 20 missions (10 real and 10 sim in the hybrid setting).

Remark: We observed that models regressing on groundtruth airspeed in simulation as opposed to noisy Pitot tube measurements provide more accurate estimates (in terms of groundtruth prediction), as expected. The simulated Pitot tube measurement includes errors such as random noise and bias. The results, from 20 training missions, are presented in Fig 4.

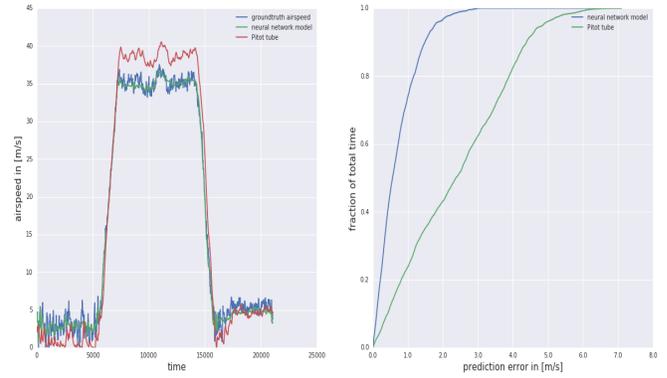


Fig. 4. Left: Pitot tube and neural network model predictions on the sim test mission. This time the neural network model was trained on data with groundtruth airspeed labels. Right: corresponding cdf curves.

2) *Robustness of Airspeed Prediction:* The neural network predictor was tested on two flights in “attitude mode” where a pilot flies the vehicle around so as to generate mission profiles explicitly outside the flight envelopes encountered during training. Despite the mismatch between training and validation data, the general profile of the airspeed curve is captured by the neural network model trained on 20 sim missions. For the first test mission, the error of the prediction is below 1.5 m/s for the **39.6178%** of the time. The average test error is: $\epsilon_{ave} = 2.04$ m/s, the mean squared test error is $\epsilon_{mse} = 5.92$ m²/s² and the L_∞ error is $\epsilon_\infty = 5.74$ m/s. For the second test mission, the error of the prediction is below 1.5 m/s for the **51.24%** of the time. The average test error is: $\epsilon_{ave} = 1.59$ m/s, the mean squared test error is $\epsilon_{mse} = 3.79$ m²/s² and the L_∞ error is $\epsilon_\infty = 4.96$ m/s. The results are presented in Fig. 5 and left subfigure of Fig. 6.

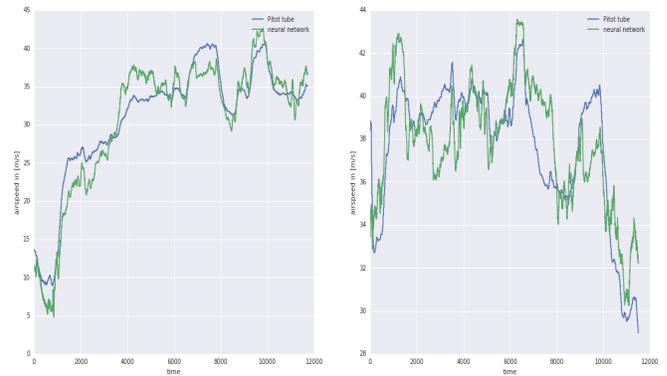


Fig. 5. Two subfigures correspond to two test flights in “attitude mode” where a pilot flies the vehicle around so as to generate mission profiles explicitly outside the flight envelopes encountered during training. The neural network model still manages to capture the shape of the airspeed curve.

Learning Aerodynamics? Does our model effectively learn correlations between actuator commands and airspeed, or does it implicitly embody aerodynamics also? We conducted experiments where we trained a model on only actuator inputs. Results are presented in right subfigure of Fig. 6. In general, while the actuators-only model is a good

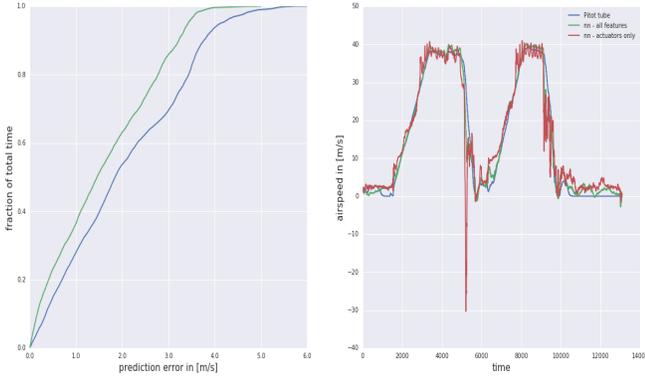


Fig. 6. Left: Cdf curves corresponding to tests from Fig. 5. Right: comparison of two neural network models: trained on all 22 features and just on actuators (10 features).

baseline, on test datasets it often produces large transient errors, e.g. see a spike in right subfigure of Fig 6). Adding 12 non-actuators features leads to smoother and more accurate predictions. Likewise, an impoverished model with no actuator commands as features performs significantly worse. In summary, using both actuator commands and vehicle kinematic state as features results in the best performance.

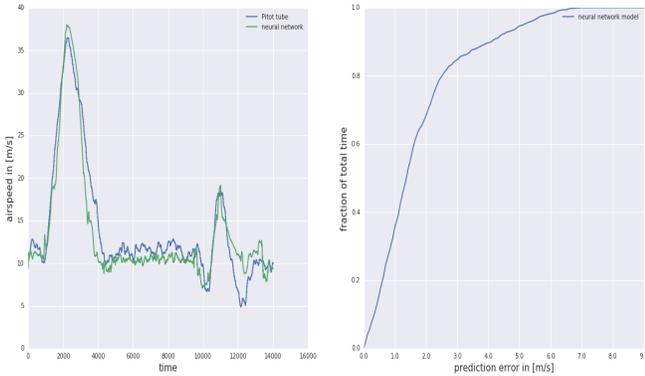


Fig. 7. On the left: Comparison of the neural network model airspeed prediction with the Pitot tube measurement. The model is trained on airspeed curves of a particular profile and tested on the other profile. On the right: the cdf curve showing what fraction of time the predicted airspeed is within ϵ range from the Pitot tube measurement for different values of ϵ .

3) Training and testing on different types of missions:

Figure 7 compares predictions made by a neural network model with Pitot tube measurements and presents a cdf curve measuring quantitatively the quality of the neural network model. Neural network models are trained on the sim data (with Pitot tube measurements as labels) with airspeed profiles characterized by two phases of relatively stable airspeeds. These two phases are connected by a short phase with airspeed changing abruptly. Test set corresponds to the mission with two "spikes" of airspeed and airspeed changing more abruptly across the entire mission.

4) *Additional experiments for Sim2Real:* We also conducted several additional experiments, where neural network model was trained on missions coming from the simulator and tested on a real missions. As datasets, we used in

particular 1000 simulated missions generated with different weather conditions (wind direction, temperature, etc.) Results are presented in the Appendix (Fig. 12 and Fig. 10).

Statistics from the test phase for the discussed above Sim2Sim and Real2Real experiments, namely: average error ϵ_{ave} , mean-squared error ϵ_{mse} and percentage of time f with error at most 1.5 m/s are presented in Fig. 8.

	ϵ_{ave} in [m/s]	ϵ_{mse} in [m ² /s ²]	f
Sim2Sim : 0	1.31617	3.04263	64.1985
Sim2Sim : 1	1.00114	1.63107	77.336
Sim2Sim : 2	1.6879	4.90287	55.0148
Sim2Sim : 3	2.43429	7.20068	18.4717
Sim2Sim : 4	2.73457	7.87346	18.21567
Sim2Sim : 5	1.90391	5.86932	53.5224
Sim2Sim : 6	1.57186	3.44613	48.1156
Sim2Sim : 7	1.15465	2.46146	73.2818
Sim2Sim : 8	1.12157	2.03307	68.1961
Sim2Sim : 9	1.84115	5.29048	49.3446
Sim2Sim : 10	1.03714	1.6679	73.5439
Sim2Sim : 11	3.07193	12.0072	19.2876
Real2Real [1, 1]	1.44312	4.47358	63.7848
Real2Real [1, 2]	1.68805	3.71016	39.4469
Real2Real [2, 1]	3.74855	15.3389	6.86966
Real2Real [2, 2]	1.16543	7.37624	86.8692
Real2Real [3, 1]	1.56724	9.21085	76.3715
Real2Real [3, 2]	0.762481	1.13623	87.7353
Real2Real [4, 1]	6.04936	83.5462	51.029
Real2Real [4, 2]	1.54749	6.16062	66.0864

Fig. 8. Statistics from the test set for Sim2Sim and Real2Real settings considered in the Appendix. Real2Real[i][j] corresponds to the scenario from the i^{th} row and j^{th} column from Fig. 11 respectively. Different Sim2Sim rows correspond to different tests.

B. Closing the Control Loop: Flying without Pitot Tubes

We integrated our proposed air data system with the on-board flight control system. The predicted airspeed then interacts with the control stack, and the whole system completes a feedback loop. Figure 9 shows a mission requiring four 180-degree turns in the presence of speed 8 m/s wind gusts, requiring the controllers to exercise the use of the airspeed for accurate vehicle guidance. We see that safe and accurate guidance and navigation is possible without Pitot tubes: the

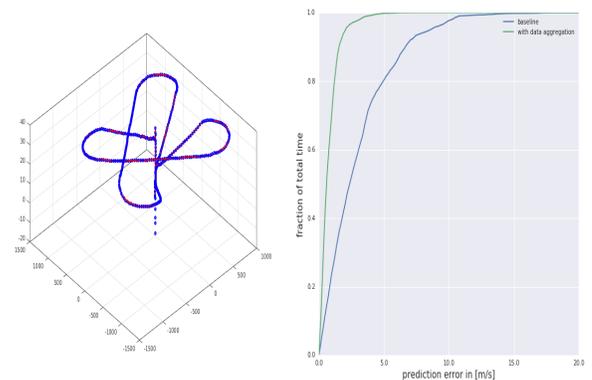


Fig. 9. Left: Flight trajectory with the neural net air data system (blue squares) closely tracks flight with pitot-tube (red). Right: Airspeed accuracy improvements with dataset aggregation.

trajectory of the Pitot-tube based flight is very accurately tracked. These results were consistent across a variety of test missions.

C. Accuracy boost from Dataset Aggregation

Here, we provide a proof of concept that model accuracy can be boosted via data aggregation techniques [7], [8] to correct potential trajectory divergence due to input distribution mismatch in feedback control loop. A baseline model trained on about 10 real missions was used to fly 20 simulated missions. The pitot tube was disconnected from the controllers (instead our neural net provided air data estimates) in these flights; however, pitot tube airspeed measurements were still recorded. Additional training data was generated by pairing sliding-window vehicle states and actuator command inputs encountered during these flights, with Pitot tube airspeeds as regression targets. The model was then retrained with this additional aggregated data. Finally, 5 novel validation missions, each approximately 4-5 minutes long were created, each requiring flight through 5 points randomly chosen on a 1 kilometer diameter circle starting and returning back the center, in the presence of 8m/s wind. On each of these validation missions, the data-aggregated model shows significantly higher quality of airspeed prediction accuracy, as summarized in Figure 9 (right).

REFERENCES

- [1] Kumar and Michael, “Opportunities and challenges with autonomous micro aerial vehicles,” in *International Symposium on Robotics Research*, 2011.
- [2] D. Floreano and R. J. Wood, “Science, technology and the future of small autonomous drones,” *Nature*, vol. 521, 2015.
- [3] A. Claesson, A. Beckman, and M. Ringh, “Time to delivery of an automated external defibrillator using a drone for simulated out-of-hospital cardiac arrests vs emergency medical services,” *Journal of American Medical Association*, 2017.
- [4] J. Das, G. Cross, C. Qu, A. Makineni, Y. M. Pratap Tokekar, and V. Kumar, “Devices, systems, and methods for automated monitoring enabling precision agriculture,” *IEEE International Conference on Automation Science and Engineering (CASE)*, 2015.
- [5] G. Gallagher, L. Higgins, L. Khinoo, and P. Pierce, *U.S. Naval Test Pilot School Flight Test Manual, NO 108, Fixed Wing Performance*.
- [6] A. Pasztor, “Faa issues safety directive concerning airspeed sensors on boeing 787 jetliners,” *Wall Street Journal*, March 31 2016.
- [7] S. Ross, G. J. Gordon, and J. A. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” *arXiv preprint arXiv:1011.0686*, 2010.
- [8] A. Venkatraman, M. Hebert, and J. A. Bagnell, “Improving multi-step prediction of learned time series models,” in *AAAI*, 2015, pp. 3024–3030.
- [9] P. Abbeel, V. Ganapathi, and A. Y. Ng, “Learning vehicular dynamics, with application to modeling helicopters,” in *Advances in Neural Information Processing Systems 18*, Y. Weiss, P. B. Schölkopf, and J. C. Platt, Eds. MIT Press, 2006, pp. 1–8.
- [10] S. Ross and D. Bagnell, “Agnostic system identification for model-based reinforcement learning,” in *ICML*, 2012.
- [11] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. Rehg, B. Boots, and E. Theodorou, “Information theoretic MPC for model-based reinforcement learning,” in *Proceedings of the 2017 IEEE Conference on Robotics and Automation (ICRA)*, 2017.
- [12] G. Kahn, T. Zhang, S. Levine, and P. Abbeel, “PLATO: policy learning using adaptive trajectory optimization,” *CoRR*, vol. abs/1603.00622, 2016.
- [13] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, “Learning monocular reactive UAV control in cluttered natural environments,” *CoRR*, vol. abs/1211.1690, 2012.
- [14] W. Sun, A. Venkatraman, B. Boots, and J. A. Bagnell, “Learning to filter with predictive state inference machines,” in *Proceedings of the 2016 International Conference on Machine Learning (ICML)*, 2016.
- [15] A. Venkatraman, W. Sun, M. Hebert, B. Boots, and J. A. Bagnell, “Inference machines for nonparametric filter learning,” in *Proceedings of the 2016 International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.
- [16] W. Sun, R. Capobianco, G. J. Gordon, J. A. Bagnell, and B. Boots, “Learning to smooth with bidirectional predictive state inference machines,” in *Proceedings of The International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016.
- [17] T. Hamel and R. Mahony, “Attitude estimation on SO(3) based on direct inertial measurements,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2170–2175.
- [18] J. Tan, Z. Xie, B. Boots, and C. K. Liu, “Simulation-based design of dynamic controllers for humanoid balancing,” in *Proceedings of The IEEE Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [19] F. Sadeghi and S. Levine, “CAD2RL: Real single-image flight without a single real image,” 2017.
- [20] J. N. Olhausen, “The use of a navigation platform for performance instrumentation on the yf-16 flight test program,” in *AIAA 13th Aerospace Sciences Meeting*, 1975.
- [21] E. Joseph, “Angle of attack and sideslip estimation using an inertial reference platform,” DTIC Document, Tech. Rep., 1988.
- [22] S. Myschik, F. Holzappel, and G. Sachs, “Low-cost sensor based integrated airdata and navigation system for general aviation aircraft,” 2008.
- [23] K. A. Wise, “Flight testing of the x-45a j-ucas computational alpha-beta system,” 2006.
- [24] C. H. Westhelle, “X-38 backup air data system (aerodad),” in *40th AIAA Aerospace Sciences Meeting and Exhibit*.
- [25] F. A. P. Lie and D. Gebre-Egziabher, “Synthetic air data system,” *Journal of Aircraft*, vol. 50, no. 4, pp. 1234–1249, 2013.
- [26] —, “Sensitivity analysis of model-based synthetic air data estimators,” in *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 0081.
- [27] M. Shaqura and C. Claudel, “A hybrid system approach to airspeed, angle of attack and sideslip estimation in unmanned aerial vehicles,” in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, June 2015, pp. 723–732.
- [28] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. John Wiley & Sons, 2015.
- [29] J. D. Anderson Jr, *Fundamentals of aerodynamics*, 3rd ed., 2001.
- [30] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, 2015.
- [31] M. Abadi *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org.

V. APPENDIX: LEARNING TO SENSE THE AIR: SELF-FLYING VEHICLES WITHOUT AIRSPEED SENSORS

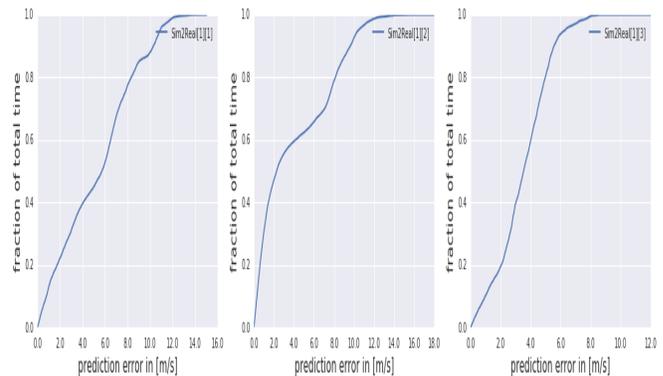


Fig. 10. Sim2Real train-test scenario: cdf curves corresponding to first three models for which test results are presented in Fig. 12.

In Figure 11 additional experiments regarding training and testing the neural network on data coming from the real missions are given. Figures 12 and 10 show sim-to-real transfer. Training was conducted on a dataset of 1000 missions from the simulator. Comparison of the airspeed measured by the Pitot tube with the one predicted by the neural network is on Fig 12 whereas the corresponding cdf curves are presented on Fig. 10.

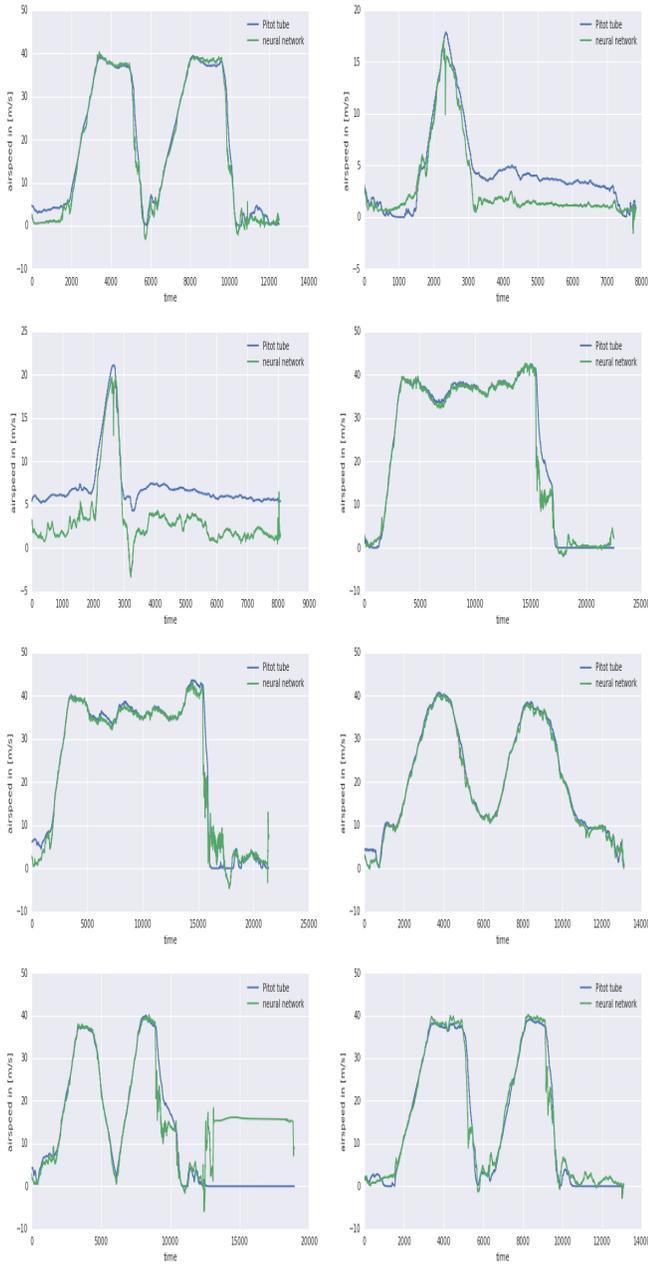


Fig. 11. Real2Real train-test scenario: Comparison of the neural network prediction with the Pitot tube measurements. Neural network models were trained on the real data, each training datasets consisted of 20 missions. The trainer used 5 epochs. Testing was conducted on another real mission.

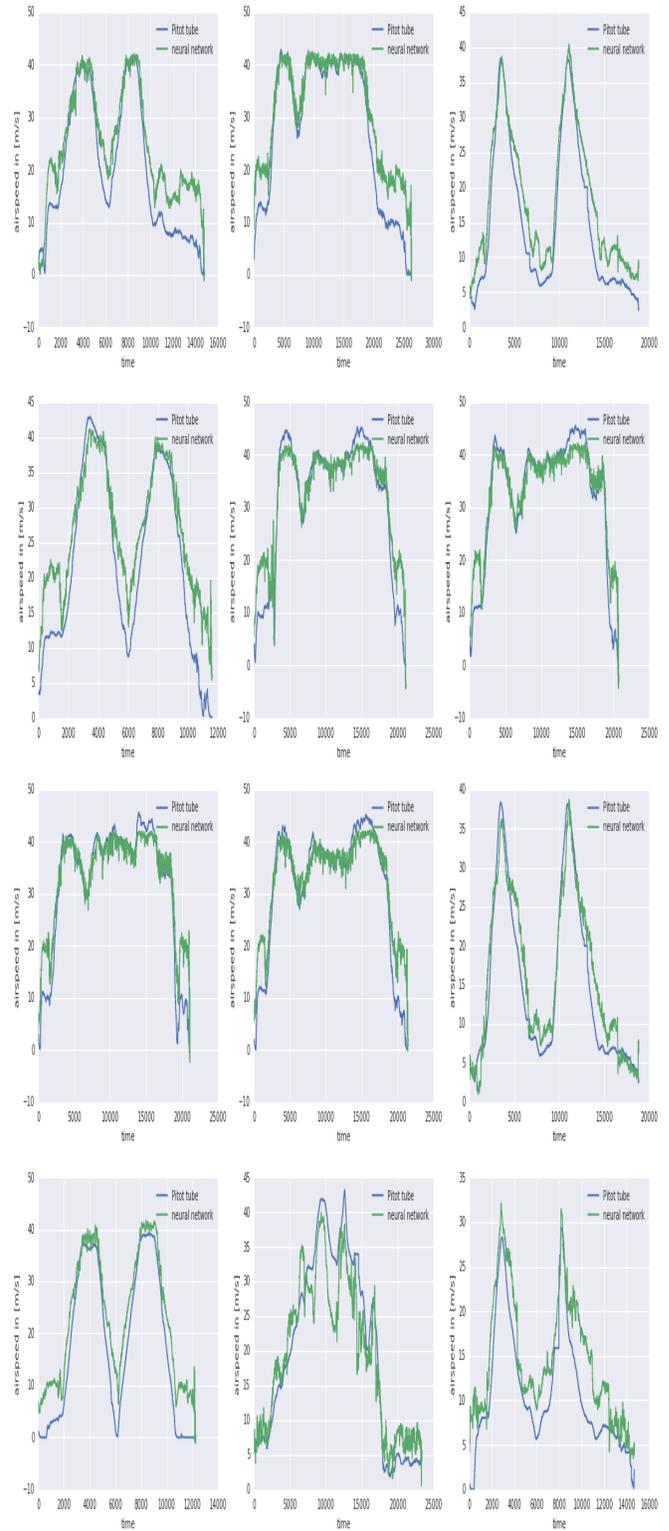


Fig. 12. Sim2Real train-test scenario: Statistics from the training phase. Neural network models were trained on the sim data. The trainer used 5 epochs. Testing was conducted on the real mission.