Deep Learning-based Flight Speed Estimation using Thermal Anemometers

Ze Wang*[1], Jingang Qu*[1,2], Pascal Morin[1] †
[1]Institut des Systemes Intelligents et de Robotique - ISIR
Sorbonne Universite, CNRS, UMR 7222, 75005 Paris, France
[2] IFPEN, France

ABSTRACT

This article concerns flight speed estimation from airflow measurements provided by a set of thermal anemometers. Our approach relies on a Gated Recurrent Unit (GRU) based deep learning approach to extract deep features from noisy and turbulent measurement signals of tri-axial thermal anemometers, in order to establish the underlying mapping between the airflow measurement and the flight speed. The proposed solution is validated on a multi-rotor micro aerial vehicle (MAV). The results show that the GRU-based model can effectively extract noise features and perform denoising, and compensate for induced velocity effects along the propellers' rotation axis. As a consequence, robust prediction of the flight speed is performed, including during takeoff and landing that induce ground effects and strong variations of vertical airflow.

SUPPLEMENTARY MATERIAL

Our open source implementation and more details are available at `https://github.com/SyRoCo-ISIR/Flight-Speed-Estimation-Airflow.git`

1 INTRODUCTION

Estimation of the 3D velocity and/or position of a MAV is essential for both tele-operated and fully autonomous missions. Different sensing modalities can be used to this purpose, which can be roughly divided into three categories: *External perception systems* (UWB odometry, GPS[1], Motion Capture system[2]), *Exteroceptive sensors* (Lidar odometry[3], visual odometry(VO)[4, 5]), and *Proprioceptive sensors* (inertial navigation system(INS)[1]). Each category of sensors comes with its own advantages and drawbacks, especially in the context of MAV applications. External perception systems, when available, are very convenient as they require limited payload (both material and computational) aboard the drone, but they are not always available.

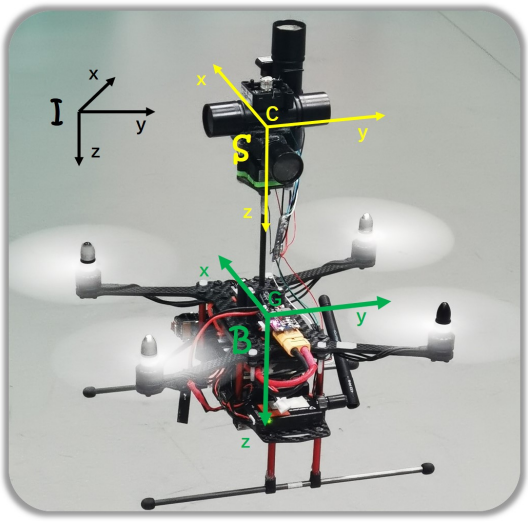

Figure 1: MAV equipped with three thermal anemometers orthogonal to each other.

Exteroceptive sensors provide information on the MAV's environment that is extremely useful for sense and avoid applications, but they often require heavy computational payload, and they are subjected to environmental conditions (light, visual texture, etc). Proprioceptive sensors (INS, or odometry based on Blade Element and Momentum theory for multi-rotor MAVs [6]) use little embarked payload, are always available, and can provide good speed estimates over limited time-intervals, but they are subjected to drift when used for position estimation. Thus, there is no universal sensory modality but each one can be useful or even crucial in specific contexts, thus contributing to a global sensor-fusion architecture.

This paper concerns the problem of velocity/air-velocity estimation from measurements provided by thermal anemometers. Here,"air-velocity" denotes the velocity relative to air. A thermal anemometer, as illustrated on Figure 1, is a cylindrical sensor (this is a typical shape, but not the only one) that provides information about the airflow that passes through the device. It can thus be viewed as a 1D air speed sensor. When used in still air (the typical indoor situation), it thus provides information on the sensor's velocity along the cylinder's axis. When used outdoor, it provides infor-

*The co-authors share the same contribution.
†This work was supported by the French ANR Project DACAR. Email addresses: {firstName.lastName}@sorbonne-universite.fr

http://www.imavs.org/

http://www.imavs.org/

mation on the sensor's air-velocity along the cylinder's axis, similar to a Pitot tube. Using a set of three orthogonal thermal anemometers to estimate the 3D velocity/air-velocity of a multi-rotor MAV, as illustrated by Figure 1, is not straightforward. The main difficulty comes from the airflow generated by the propellers, especially in the vertical direction (in body frame) where airflow velocity is the sum of the induced velocity and MAV's velocity. In addition, as with any sensor, measurements are noisy, where noise may be electronic or "mechanical" (related, e.g., to vibrations aboard the MAV). By combining a simple analytical sensor model and a neural network architecture, we show that good speed estimates can be obtained in all flight phases, including take-off and landing. In particular, the neural network proves helpful in correcting induced velocity related biases in the (body relative) vertical direction and reducing "noise". Concerning noise reduction, note that similar conclusions about the benefit of neural networks were drawn for attitude estimation based on IMU measurements [7]. We investigated several different neural networks architectures such as LSTM, MLP, and WaveNet, and found that the GRU-based architecture performs best in estimating the flight speed.

The rest of paper is organized as follows. Section 2 provides a literature review concerning the different types of airflow sensors in the context of MAVs applications. A simple model of the thermal anemometer measurement is proposed in Section 3. Outputs of this model are then used as inputs of the deep learning-based architecture presented in Section 4. Experimental results obtained with the proposed method and the MAV of Figure 1 are reported in Section 5. The paper ends with a conclusion and perspectives.

2 AIRFLOW SENSORS FOR MAVS

There are many types of anemometers that rely on different measurement techniques, and they are usually divided into two categories: *velocity anemometers*, which directly convert airflow speed into other physical signals (such as Cup/Vane anemometers, Hot-wire anemometers based on Thermal Dissipation [8, 9, 10], Laser anemometers based on Doppler shift, Ultrasonic anemometers based on ToF[11]), and *pressure anemometers*, which sense pressure differences and thus derive speed according to Bernoulli's principle (such as Pitot-Tube anemometers [12]).

Recently, researchers have also designed custom-made sensors based on the above principles. Inspired by bionic ideas, a whisker-like sensor is presented in [13], and combined with a barometer that acts as a force sensor to measure the airflow pressure signal coming from the whiskers. The idea of a bionic whisker is also employed in [14] to design a dedicated anemometer but Hall sensors are used to measure the displacement of magnetic objects caused by air drag. Another work that also uses Hall sensors is [15], but differently it leverages airflow pressure to push a pendulum-like plate, thus using Hall sensors to measure the angle of deflection and thus predict the airflow speed. Other solutions based on converting the airflow information into a motion information include [13, 14, 15], where both [14] and [15] use Hall sensors to directly capture the electrical signal from the motion, while [13] converts it into a pressure variation on an embedded barometer. In comparison, the whisker-like structure has a smaller size and thus is more sensitive to the variation of airflow and thus has a higher resolution, while the plate structure has a larger measurement range and less turbulence effects. The work [14] stands perfectly in-between: wide measuring range on the one hand and higher resolution on the other hand. All these solutions rely and mechanical moving parts, however, which can be an issue in term of mechanical robustness and maintenance.

Solutions that do not require mechanical moving parts include laser Doppler anemometers, ultrasonic anemometers, thermal anemometers, and Pitot tubes. Laser Doppler anemometers are suitable for high precision scenarios but are not suited to MAVs due to their size and cost. Pitot tubes are unidirectional and usually applied to fixed-wing aircraft. The TriSonica Mini ultrasound anemometer was used in [11]. Although being the most compact 2D ultrasound anemometer on the market, it is still heavy for MAV applications and expensive. The SFM3000 thermal anemometer was used in both [10] and [9]. It is a 1D bidirectional anemometer weighing 17g, with a data rate of 2kHz, a measurement range of 200 standard liters per minute (slm) and a resolution of ±2.5%. This sensor was also chosen for this work due to its numerous advantages: low pressure loss, large flow range, high accuracy, high repeatability, high reliability, good linearity and sensitivity, and easy maintenance due to the absence of moving mechanical parts.

3 THERMAL ANEMOMETER MEASUREMENT MODEL

Bi-directional thermal anemometers[16, 17], measure the airflow speed by monitoring the amount of heat dissipated from a surface using one or more simple temperature sensors. For such a sensor, the 1D bi-directional measurement $m \in \mathbb{R}$ depends on the relative air velocity $v_a \in \mathbb{R}$ along the measurement axis of the sensor, where v_a is the difference between the sensor's linear velocity and the ambient wind velocity, both projected along the measurement axis of the sensor. We model the measurement by assuming a Gaussian stochastic white noise.

$$m = f(v_a) + \varepsilon \ , \quad \varepsilon \sim N(0, \sigma^2) \tag{1}$$

where $f(\cdot)$ is a strictly monotonic function whose inverse function is $g(\cdot)$ and ε is a noise which depends on the sensor characteristics and the turbulence.

3.1 Sliding window average denoising

Thermal anemometers measurement noise usually depends on the characteristics of the sensor itself and the complex turbulence around the sensor. Usually, the circuit noise

http://www.imavs.org/

and vibration has homoscedasticity characteristics (assumption of equal or similar variances over the entire period) and the complex turbulence can also be regarded as having local homoscedasticity, (e.g. Figure 3, Figure 5). For such signals, sliding window average denoising method is the most easily used to filter out noise. Consequently, the noise-filtered sensor measurement model can be written

$$\hat{m} = f(v_a) \tag{2}$$

3.2 Linear model

A commercially available anemometer usually has a reasonable measurement linearity, so we consider the approximation function as a biased linear function.

$$v_a = \frac{\hat{m} - b}{s} \in \mathbb{R} \tag{3}$$

where b is the bias and s is the scale, which need to be identified from the filtered anemometer measurements $\hat{m}$ and ground truth measurements of v_a. After combining many sampling data, we compose them in the form of a matrix:

$$[v_a] = AX(\hat{m}), \quad A = (-b/s, 1/s) \tag{4}$$

where $[v_a] = (v_a(t_1) \ \cdots \ v_a(t_N))$ and

$$X(\hat{m}) = \begin{pmatrix} 1 & \cdots & 1 \\ \hat{m}(t_1) & \cdots & \hat{m}(t_N) \end{pmatrix}$$

contain ground truth air velocity measurements and anemometer measurements at different sampling times $t_1, \cdots, t_N$, respectively. Ground truth air velocity measurements are provided by a SLAM system, as discussed later in Section 5. Minimizing the squared sum of estimation errors

$$\|[v_a] - AX(\hat{m}))\|^2 \tag{5}$$

gives the least squares estimator

$$\hat{A} = [v_a]X(\hat{m})^T(X(\hat{m})X(\hat{m})^T)^{-1} \tag{6}$$

After identification of A, the air velocity estimate associated with a measurement m is thus given by: $\hat{v}_a = \hat{g}(m) = \hat{A}(1, m)^T$.

3.3 3D triaxial anemometer sensor

In this work, we built a triaxial sensor composed of three anemometers mounted in an orthogonal configuration, as shown on Figure 1. The objective is to estimate the 3D velocity/air-velocity. The following notation is used:

- $\mathcal{I}$ is an inertial frame.
- $\mathcal{B}$ is a body frame (i.e., attached to the MAV), with center at the MAV's center of mass G. The rotation matrix from $\mathcal{B}$ to $\mathcal{I}$ is denoted as R and it satisfies the equation $\dot{R} = R[\omega]_\times$ with ω the angular velocity vector expressed in body frame and $[\omega]_\times$ the skew-symmetric matrix associated with the cross product by ω.
- $\mathcal{S}$ is a sensor frame (i.e., attached to the triaxial anemometer sensor), with center at the sensor's center C. The constant rotation matrix from $\mathcal{S}$ to $\mathcal{B}$ is denoted as R_0 and the vector of coordinates of C in the body frame $\mathcal{B}$ is denoted as δ.
- V^G denotes the coordinates, in inertial frame, of the velocity vector of G (i.e., the MAV's velocity vector).
- V_w denotes the coordinates, in inertial frame, of the wind speed vector.

The objective is to estimate the MAV's air-velocity vector $V_a^G = V^G - V_w$.

Let V^C denote the coordinates, in inertial frame, of the velocity vector of C. We have $V^C = V^G + R(\omega \times \delta)$ and thus

$$V_a^C := V^C - V_w = V_a^G + R(\omega \times \delta)$$

where V_a^C denotes the air-velocity vector at the sensor center C, expressed in inertial frame. Using the linear model proposed in Section 3.2, the triaxial anemometer sensor provides the components of this vector in the sensor frame $\mathcal{S}$. Since the rotation matrix from sensor frame to inertial frame is $\bar{R} = RR_0$, the vector of sensor measurements is

$$\begin{aligned} \bar{V}_a^C = \bar{R}^T V_a^C &= \bar{R}^T V_a^G + \bar{R}^T R(\omega \times \delta) \\ &= R_0^T \left(R^T V_a^G + \omega \times \delta\right) \end{aligned}$$

which is equivalent to

$$R^T V_a^G = R_0 \bar{V}_a^C - \omega \times \delta \tag{7}$$

In the following section we propose a deep learning-based solution to estimate $R^T V_a^G$ from the vector of triaxial sensor measurements $\bar{V}_a^C$ and the gyroscope measurement ω. The constant parameters R_0 and δ in the above relation will be implicitly estimated by the neural network. For the MAV used in this work, R_0 is probably close to the identity matrix but no calibration was performed.

4 DEEP LEARNING-BASED PREDICTION WITH AIRFLOW

Model-based and data-driven solutions are two different state-estimation methods, each of which has different advantages. The model-based approach usually employs Kalman filters [11, 15] to fuse multiple sensor data, low-pass filters [10] to reduce noise, or construct dynamical models [2] to extract airflow speed terms by referring to other measurements, respectively. A data regression method is employed in [9], and a deep learning-based method is employed in [18]. Kalman-based filters are typically the most efficient and reliable for fusing multi-sensor data, while deep learning-based methods have better performance for extracting data features, denoising, and data regression. Turbulence causes heteroscedastic disturbance noise (e.g in Figure 3), which calls for data-driven approaches to identify temporary noise

characteristics. In addition, the induced flow on multi-rotor MAVs leads to regular coupling of measurement data from anemometers in different directions. Therefore in this section we present our deep learning-based method, which is able to learn local noise features and thus perform denoising, compensate for unknown constant parameters (R_0 and δ), and decouple the nonlinear relationship between data by deep learning a large amount of measurement data. We use a neural network composed of 1D convolution layers and Gated Recurrent Unit (GRU) layers [19].

4.1 *Inputs & Outputs*

Due to the difficulty to obtain ground truth air velocity measurements, experiments are performed indoor, under the assumption of no ambient wind. Thus the MAV's velocity is equal to its air velocity. The velocity ground truth is obtained from a Visual-SLAM. According to (7), the inputs consist of the air velocity deduced from the raw anemometer measurements and the linear model of Section 3.2, and the angular velocity measurements. Note that we have tried to add accelerometer measurements and motor commands as input of the neural network, but it has led to over-fitting. Inputs (resp. outputs) of the network are expressed in the sensor frame (resp. body frame) in order to keep the network invariant with respect to the orientation of vehicle. This is important in order to reduce the complexity of the network and the amount of training data.

4.2 *Neural Network Architecture*

The architecture of the neural network is shown on Figure 2. The "Airflow-sensor data" on this figure correspond to the airspeed estimates obtained from the linear model of Section 3.2 using raw (i.e., non-filtered) anemometer measurements. Note, however, that for the linear model identification, i.e., linear regression, anemometer measurement filtering has been performed. The airflow-sensor data, together with the raw gyroscope measurements, are first processed by the scaling layer, which normalizes the inputs. Then, to denoise the inputs, we use two 1D convolution layers, for which the number of filters is 16, the kernel size is 5, the stride is 1, the activation function is the rectified linear unit (ReLU)[20], and the input and output sequences are of the same length by padding with zeros. Finally, two GRU layers with 16 units are stacked on top of convolution layers to handle sequences. Compared to Long-Short Term Memory (LSTM) [21], GRU has fewer learnable parameters and thus less risk of over-fitting. Eventually, a fully connected output layer is used to integrate the deeply processed information to produce the final flight speed prediction. The neural network is built upon TensorFlow[22].

5 IMPLEMENTATION & RESULTS

5.1 *Implementation Details*

We collect data on the custom-built MAV flight platform of Figure 1 with length x width x height equal to 15x20x20cm and weight about 500g. The MAV is equipped with Pixhawk 4 open source flight control board (STM32f7), D435i binocular depth camera, and Khadas Vim 3 SBC (6-core ARM 2.4GHz).

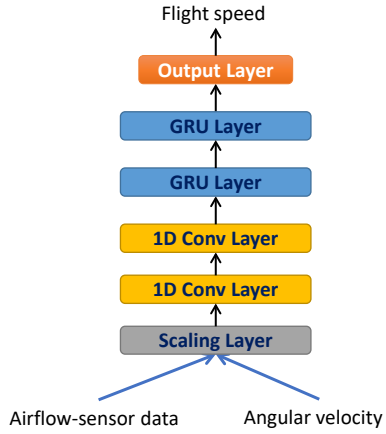

Figure 2: Neural network architecture

The open source flight control system PX4 runs on Pixhawk 4 and contains a 4-loops cascade PID controller, an Extended Kalman Filter (EKF) based state estimator, and uses the mavlink protocol to communicate with Vim 3 via the serial port. All of the components run in a ROS system on Vim 3 and communicate with an off-board laptop via wifi 5. The ground truth velocity is provided by the EKF based on the ORB-SLAM3[5] output and the IMU measurements. The EKF relies on the accelerometer measurement model.

The thermal mass flow sensors we use are the SFM3000-200 from Sensirion AG. The sampling rate of 200Hz here used and the high speed I2C communication interface make it easy to acquire data. All of our flight data has been collected in manual flight or automatic trajectory tracking in a standard gymnasium on different days.

A trajectory generator ROS node commands the vehicle to follow a trajectory in flight, including a circular spiral and a figure-of-8 spiral with a radius of 2 meters and an undulation height of 1 meter. The takeoff operation is also commanded by the ROS node during the automatic trajectory tracking.

The low priority of the data logging, due to the limitations of the autopilot hardware, would lose data randomly at different moments, but the total lost data would not exceed 2% of the total. For this reason, we compensated for the missing ground truth data using a sliding window average cubic polynomial fitting method, while we approximated the missing raw sensor measurements using a linear interpolation method. We do not have statistics of slam accuracy, but the comparison [4] shows that orb-slam3 exhibits performances similar to other state-of-the-art slam methods.

5.2 *Neural network training*

We divide the collected data into the training set (four flights data with a total of 20.18 minutes), validation set (two

http://www.imavs.org/

Table 1: Parameters identification with linear regression

Parameters	Sensor-F	Sensor-R	Sensor-D
Bias b	-0.0020	0.0154	-0.2143
Scale s	0.3100	0.2984	0.1420

flights data with 14.47 minutes), and test set (one flight data with 4.04 minutes). To compensate for the differences in flights data on different days (air temperature, humidity, pressure, etc. on the day), both the training data-set and the validation data-set contain multiple flight data from different days. The batches of sequences are created by sliding a 2.5 seconds time window which contains 500 data. The batch size has been set as 512, and we use the Adam with Triangular Cyclic Learning Rate (CLR)[23] as the optimizer, which periodically increases and decreases the learning rate during the training. Early stopping is also used to avoid over-fitting.

5.3 Anemometer measurement noise characteristics

Figure 3 displays air-velocity estimates along the sensor front direction obtained with the linear model, before filtering (blue) and after filtering (red). The flight is composed of five different flight phases performed in automatic flight mode: takeoff, circular spiral trajectory, figure-of-eight spiral trajectory, vertical flight, and landing. The trajectory plots are shown in Figure 4, where the blue curve is the trajectory and the gray curves are its projection onto the x, y, and z axes. We have used here a sliding window of length 19 to generate delay-free smooth data and obtains better noise characteristics. Of course, this sliding window is not used online for the airspeed estimation. The error statistics are presented in Figure 5, which shows that the anemometer measurements present a Gaussian white noise characteristic in the whole and Figure 3 shows that the measurements locally present a disturbance characteristic related to the flight state.

5.4 Prediction of linear model and GRU model

For the test flight, the trajectory of Figure 4 was used. The prediction results are shown in Figure 6, where we compare the flight speed prediction of the linear model of Section 3.2 (green dashed line) with the flight speed prediction of the GRU model (blue dashed line) and the ground truth flight speed (red solid line).

The parameters of the identified linear model are presented in Table.1, where Sensor-F, Sensor-R, and Sensor-D stand for the Front, Right, Down directions in the sensor frame respectively. As expected from the symmetry of the system, parameters for the Sensor-F and Sensor-R are very close. Concerning the vertical direction (i.e., Sensor-D), we identified this model using only flight data from hovering and vertical flights, so as to avoid the impact of horizontal flight velocity on the induced velocity. When the horizontal flight is dominant, the linear model estimation in the F-R direction shows essentially the consistency with the ground truth, while the resulting induced flow causes bias in the D-direction estimation. When vertical flight is dominant, the air is more likely to be accelerated downward in comparison with the drone's body and thereby change the induced flow speed faster than flight speed, so that the linear model estimation in D-direction is ahead of the true flight speed. The vertical flight speed also shows a slight coupling characteristic in the horizontal measurements.

Figure 7 shows the flight speed prediction error of the linear model w.r.t the ground truth versus the flight speed prediction error of the GRU model w.r.t the ground truth in each direction. The variances of the GRU model in each direction are 0.0111, 0.0132, 0.0143, while that of the linear model in the F-R direction are 0.0421, 0.0377, respectively.

We summarize as follows: (1) The prediction results of GRU largely overlap with the ground truth, better than linear model, which indicates that the neural network successfully discovers patterns in the data. (2) The GRU model flight speed prediction curve is smooth, which indicates that the neural network identifies the noisy feature and thus performs denoising process. (3) The anemometer measurements in the D-direction implicitly contain the induced flow speed, which is coupled with the flight state, and which is usually difficult to predict and the linear model performs poorly for the prediction of the flight speed in trajectory tracking. The GRU model, however, shows better performance, especially during the takeoff and landing phases. This implies that the GRU model is very robust. (4) For the flight speed prediction in the F-R direction, both the linear model and the GRU model match the ground truth well but Figure 7 shows that the GRU model largely outperforms the linear model.

6 CONCLUSIONS & FUTURE WORK

In this work, we addressed the velocity/air velocity estimation problem aboard MAVs based on thermal anemometer measurements. The proposed solution uses a set of three sensors mounted in a triaxial orthogonal configuration and combines a simple analytical model with a neural network. By performing all computations in body and sensor frames, the problem is invariant with respect to rotations, which reduces the complexity of the neural network and the amount of required training data. Validations show that the neural network brings significant benefits in term of denoising and compensation of induced-flow.

Perspectives include using other sensors available onboard the MAV, such as accelerometer barometer, to further improve speed prediction by using a Kalman method, or exploiting anemometer measurements for obstacles detection.

ACKNOWLEDGMENTS

The first author is funded by the China Scholarship Council (CSC) from the Ministry of Education of P.R. China. Thank to Prometheus platform [24] support.

http://www.imavs.org/

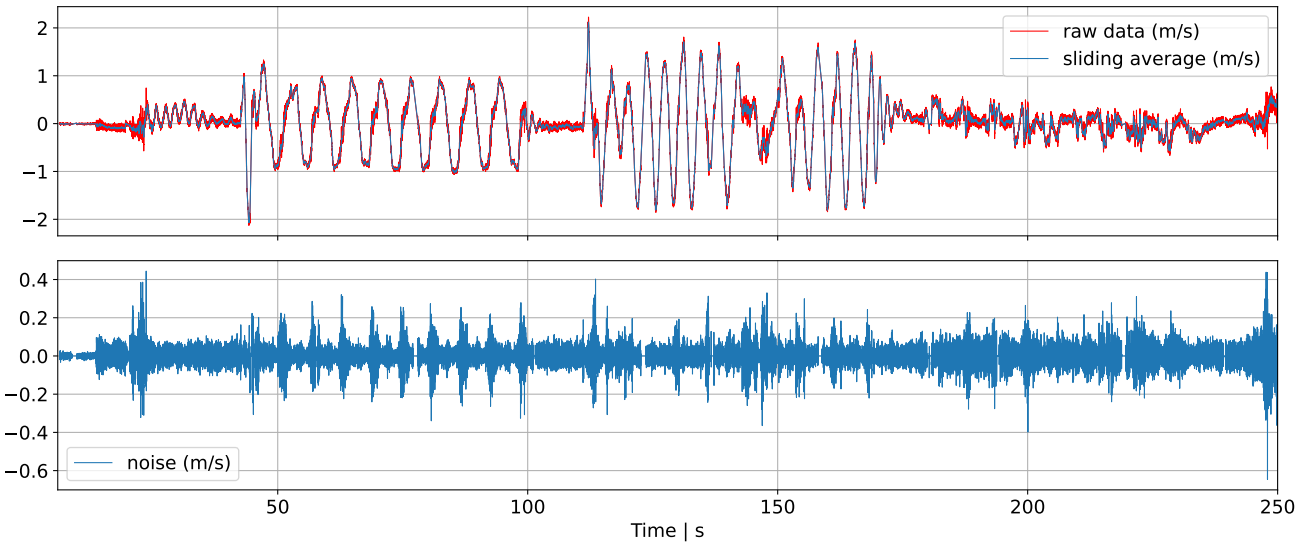

Figure 3: Sliding window average denoise. [Top] The air-velocity estimate with the linear model in the Sensor Front direction before filtering (blue) and after filtering (red). [Bottom] The filtered noise shows that it presents almost homoscedasticity over the entire period, while locally, especially when the flight speed direction changes, it presents heteroscedasticity from the whole.

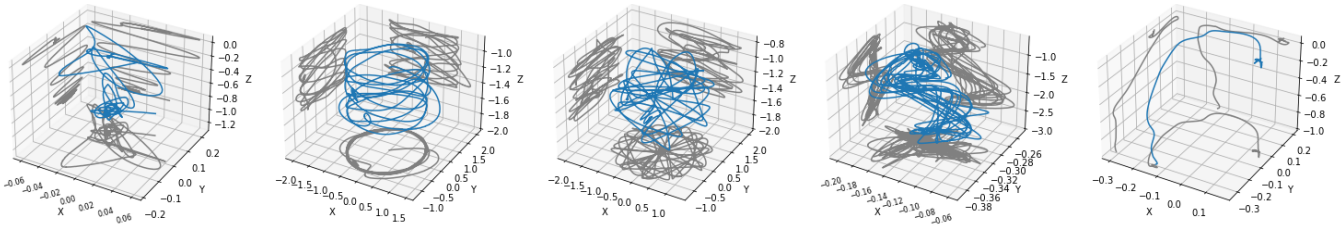

Figure 4: The plots show takeoff, circle spiral, figure-of-8, vertical flight, and landing in order from left to right.

http://www.imavs.org/

REFERENCES

[1] AH Mohamed and KP Schwarz. Adaptive kalman filtering for ins/gps. *Journal of geodesy*, 73(4):193–203, 1999.

[2] LNC Sikkel, Guido CHE de Croon, Christophe De Wagter, and QP Chu. A novel online model-based wind estimation approach for quadrotor micro air vehicles using low cost mems imus. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2141–2146. IEEE, 2016.

[3] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[4] Dinar Sharafutdinov, Mark Griguletskii, Pavel Kopanev, Mikhail Kurenkov, Gonzalo Ferrer, Aleksey Burkov, Aleksei Gonnochenko, and Dzmitry Tsetserukou. Comparison of modern open-source visual slam approaches. *arXiv preprint arXiv:2108.01654*, 2021.

[5] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[6] Alexandre Letalenet and Pascal Morin. Identification and evaluation of a force model for multirotor uavs. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4280–4286. IEEE, 2020.

[7] Martin Brossard, Silvere Bonnabel, and Axel Barrau. Denoising imu gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robotics and Automation Letters*, 5(3):4796–4803, 2020.

[8] Paolo Bruschi, Massimo Piotto, F Dell'Agnello, J Ware, and N Roy. Wind speed and direction detection by means of solid-state anemometers embedded on small quadcopters. *Procedia Engineering*, 168:802–805, 2016.

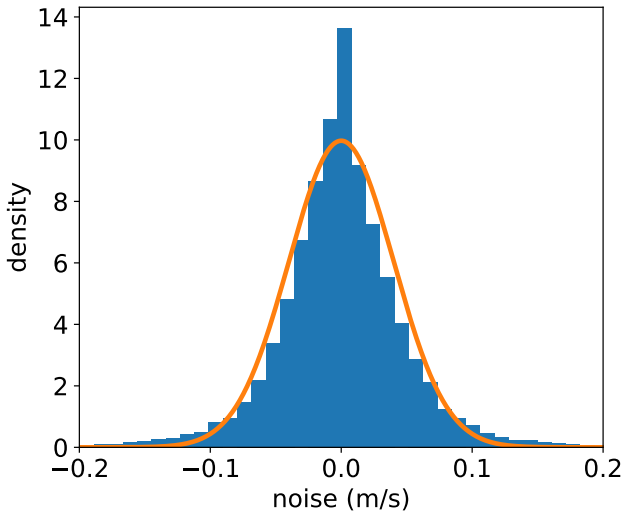

Figure 5: Histogram of the statistical distribution of the filtered noise. Generally, the noise presents Gaussian white noise characteristics with the mean of $2.5 \times 10^{-3} m/s$ and the std of $0.04 m/s$.

[9] Jörg Müller, Oliver Paul, and Wolfram Burgard. Probabilistic velocity estimation for autonomous miniature airships using thermal air flow sensors. In *2012 IEEE International Conference on Robotics and Automation*, pages 39–44. IEEE, 2012.

[10] You Li, Shady Zahran, Yuan Zhuang, Zhouzheng Gao, Yiran Luo, Zhe He, Ling Pei, Ruizhi Chen, and Naser El-Sheimy. Imu/magnetometer/barometer/mass-flow sensor integrated indoor quadrotor uav localization with robust velocity updates. *Remote Sensing*, 11(7):838, 2019.

[11] Derek Hollenbeck, Gregorio Nunez, Lance E Christensen, and YangQuan Chen. Wind measurement and estimation with small unmanned aerial systems (suas) using on-board mini ultrasonic anemometers. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 285–292. IEEE, 2018.

[12] Sam Prudden, Alex Fisher, Matthew Marino, Abdulghani Mohamed, Simon Watkins, and Graham Wild. Measuring wind with small unmanned aircraft systems. *Journal of Wind Engineering and Industrial Aerodynamics*, 176:197–210, 2018.

[13] William Deer and Pauline EI Pounds. Lightweight whiskers for contact, pre-contact, and fluid velocity sensing. *IEEE Robotics and Automation Letters*, 4(2):1978–1984, 2019.

[14] Andrea Tagliabue, Aleix Paris, Suhan Kim, Regan Kubicek, Sarah Bergbreiter, and Jonathan P How. Touch the wind: Simultaneous airflow, drag and interaction sensing on a multirotor. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1645–1652. IEEE, 2020.

[15] Shady Zahran, Adel M Moussa, Abu B Sesay, and Naser El-Sheimy. A new velocity meter based on hall effect sensors for uav indoor navigation. *IEEE Sensors Journal*, 19(8):3067–3076, 2018.

[16] Oleg Sazhin. Novel mass air flow meter for automobile industry based on thermal flow microsensor. i. analytical model and microsensor. *Flow Measurement and Instrumentation*, 30:60–65, 2013.

[17] Doug Sparks. Mems pressure and flow sensors for automotive engine management and aerospace applications. *MEMS for Automotive and Aerospace Applications*, pages 78–105, 2013.

[18] Andrea Tagliabue and Jonathan P How. Airflow-inertial odometry for resilient state estimation on multirotors. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5736–5743. IEEE, 2021.

[19] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[20] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[22] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[23] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.

[24] Prometheus Autonomous drone open source project. `https://github.com/amov-lab/Prometheus/`.

http://www.imavs.org/

http://www.imavs.org/

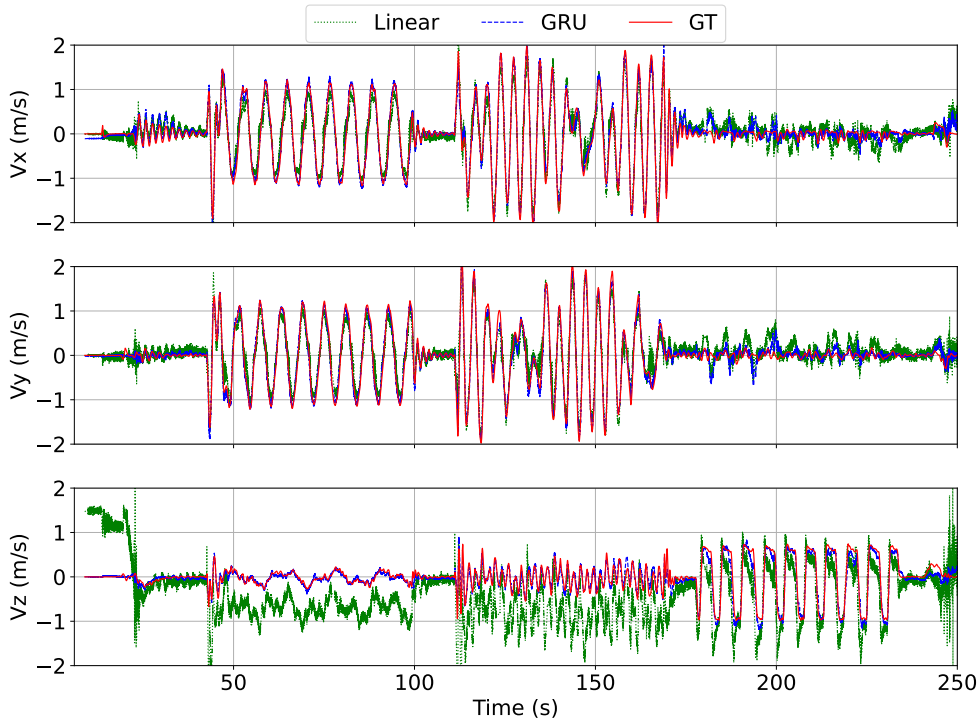

Figure 6: Five flight phases are respectively in chronological order the take-off (<30s), the circular spiral trajectory (40s-100s), figure-of-eight trajectory (120s-170s), vertical flight (180s-230s), and the landing (240s-250s). The flight speed prediction of the identified linear model without removing the noise (green) vs. that of the learning-based model (blue) vs. the ground truth flight speed from V-SLAM (red). All is expressed in Body-FRD frame.

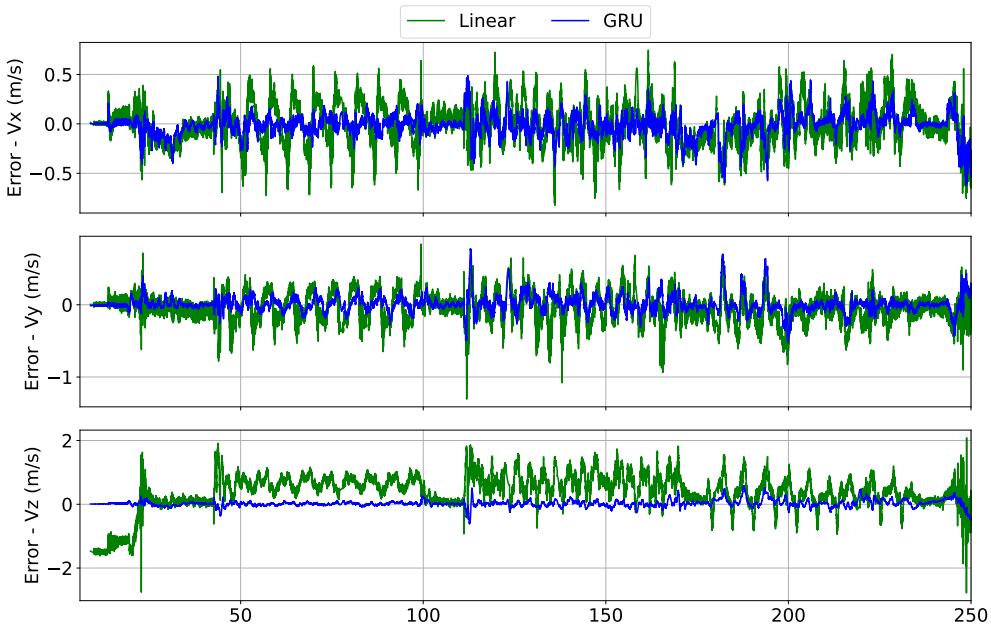

Figure 7: The flight speed prediction error of linear model w.r.t the ground truth (green) vs. the flight speed prediction error of deep learning-based model w.r.t the ground truth (blue). The variances of the GRU model in the each direction are 0.0134, 0.0133, 0.0139, while the variances of the linear model in the F-R direction are 0.0421, 0.0377, respectively.