Lumped Drag Model Identification and Real-Time External Force Detection for Rotary-Wing Micro Aerial Vehicles

Lucas Wälti and Alcherio Martinoli

***Abstract*— This work focuses on understanding and identifying the drag forces applied to a rotary-wing Micro Aerial Vehicle (MAV). We propose a lumped drag model that concisely describes the aerodynamical forces the MAV is subject to, with a minimal set of parameters. We only rely on commonly available sensor information onboard a MAV, such as accelerometer data, pose estimate, and throttle commands, which makes our method generally applicable. The identification uses an offline gradient-based method on flight data collected over specially designed trajectories. The identified model allows us to predict the aerodynamical forces experienced by the aircraft due to its own motion in real-time and, therefore, will be useful to distinguish them from external perturbations, such as wind or physical contact with the environment. The results show that we are able to identify the drag coefficients of a rotary-wing MAV through onboard flight data and observe the close correlation between the motion of the MAV, the measured external forces, and the predicted drag forces.**

I. INTRODUCTION

Rotary-wing Micro Aerial Vehicles (MAVs) and, more generally, Unmanned Aerial Vehicles (UAVs) are used in a multitude of fields today. Their application includes crop and environmental monitoring [1], [2], imagery and photogrammetry [3], infrastructural asset inspection [4]–[7], or search and rescue applications [8]. However, an outdoor-operating MAV will likely be subject to disturbances, mostly wind-induced, which can increase the risk of failure during a mission. Therefore, the detection of disturbances is important from a stabilization and control standpoint.

An intuitive way to perform wind sensing is through the use of dedicated sensors [1], [9], [10]. However, the addition of additional sensors reduces the payload that the aircraft can carry and has a negative impact on the autonomy of the aircraft. Since wind sensors have to be placed far enough from the rotors to avoid being influenced by their wake, the footprint of the vehicle increases, which is not desirable for aircrafts engaged in applications that include flying close to infrastructures or within confined spaces. Instead, model-based solutions have been investigated to avoid the use of dedicated sensors. The simplest approach consists of observing the attitude of the aircraft in hovering conditions and linking the observed attitude with the intensity of the wind [1], [11], [12]. However, this approach significantly limits the motion of the aircraft. To overcome these limitations, approaches have been proposed for dynamic flights. Although the thrust exerted by the system can be deduced from the hovering conditions for stationary flights, it needs to be directly measured in dynamic flights, as the only two types of force exerted on a rotary-wing aircraft are the thrust and aerodynamical forces, either induced by its own motion or external wind. In the literature, it is common to use direct measurements of the generated thrust by measuring either the velocity of the rotors directly or the current drawn by the Electronic Speed Controllers (ESCs) [11], [13]–[17]. This approach requires low-level integration with the system, which may not always be possible. Otherwise, encoders or Hall-effect sensors can be added to the motors at the cost of additional weight and hardware complexity. To mitigate this, a low bandwidth assumption on the wind profile has been proposed in [18], [19], which makes it possible to reuse the previous wind estimate to predict the current thrust and, therefore, estimate the new wind conditions. However, this reduces the applicability of the approach, since assumptions about the wind profile are made. While this solution could be suitable in open-field applications where the general wind conditions must be observed (e.g., in the scope of meteorological studies), its application is not suitable where more dynamic conditions apply.

Regardless of the selected approach, the aerodynamic characteristics of the aircraft need to be identified. They are usually described by an aerodynamic drag model of varying complexity depending on the application [20]–[24]. Wind tunnel experiments are often performed to identify the model parameters [1], [12], [15]. However, the requirement of conducting wind tunnel experiments can be a significant limiting factor in the general applicability of a method, as such facilities are not commonly available. To avoid this limitation, several approaches have been suggested to learn the complex aerodynamic forces exerted on the aircraft from flight data through various machine-learning techniques, such as the Gaussian Process [25] or Deep Learning [26]. This provides an abstraction for the drag model capable of capturing complex interactions at the cost of using more complex and computationally demanding algorithms.

This work provides a middle ground to the solutions previously cited. We reduce the drag model to a lumped minimal form that captures the essential dynamics at play and only requires the identification of a minimal set of parameters. The thrust is deduced from the throttle commands through

Both authors are with the Distributed Intelligent Systems and Algorithms Laboratory (DISAL), School of Architecture, Civil and Environmental Engineering (ENAC), Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland. Corresponding e-mail: lucas.walti@epfl.ch

This work was partially funded by the ENAC Interdisciplinary Cluster Grant "ECOLE: IntElligent Systems for Automated InspeCtiOn of SteeL InfrastructurE".

Additional information about the research and an accompanying video are available here: https://www.epfl.ch/labs/disal/research/aerialdistributedinspection/

a model detailed later in this work. It is thus important to note that we do not require any hardware modifications or sensor integration for our solution to work, as we only require data commonly available on any rotary-wing MAV. In fact, our method only requires the MAV to provide a pose estimate (e.g., obtained from the fusion of inertial and GNSS measurements, Visual Inertial Odometry (VIO), or through another approach), accelerometer measurements, and throttle commands. In our case, the state estimate is obtained from a VIO pipeline. We show that this information suffices for the direct measurement of external forces. It can be further leveraged to identify the aforementioned lumped drag model, which allows us to predict the aerodynamical forces experienced by the MAV due to its own motion in real-time and, therefore, distinguish them from external perturbations, such as wind or physical contact with the environment.

II. PRELIMINARIES

This section presents the notation used in this work and reviews some core concepts of the dynamics of a rotary-wing MAV.

A. Notation and Conventions

The following uses the notation ${}^{\mathscr{W}}(\cdot)$ and ${}^{\mathscr{B}}(\cdot)$ to denote whether a quantity $(\cdot)$ is expressed in the world frame $\mathscr{W}$ or the body frame $\mathscr{B}$, respectively, as illustrated in Fig. 1. Vectors (lower case) and matrices (upper case) are denoted by a bold font. The element-wise multiplication is denoted by $\odot$ and the element-wise absolute value by $|\,|$. The frame convention used in this work is Forward Left Up (FLU), or equivalently North West Up (NWU). We use Tait-Bryan angles with the intrinsic rotation sequence $z-y'-x''$ (i.e. yaw, pitch, roll, denoted by the angles ψ, θ, ϕ). The attitude of the drone is generally represented by the rotation matrix ${}^{\mathscr{B}}_{\mathscr{W}}\mathbf{R}$. The total inclination is denoted by $\Theta = \arccos\left(\mathbf{u}_z^T\, {}^{\mathscr{B}}_{\mathscr{W}}\mathbf{R}\mathbf{u}_z\right)$, with $\mathbf{u}_z = [0\,0\,1]^T$.

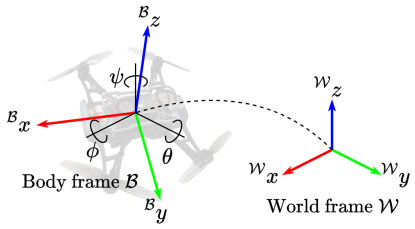

Fig. 1: Inertial world frame $\mathscr{W}$ and body frame $\mathscr{B}$, with roll, pitch and yaw angles denoted by ϕ, θ and ψ.

B. General Dynamics

The dynamics of a flying MAV can be summed very concisely with the following expression, expressed in the body frame $\mathscr{B}$:

$$m\mathbf{a} = \boldsymbol{\tau} + \mathbf{f}_{\text{ext}} \tag{1}$$

where m is the mass of the aircraft, $\mathbf{a}$ is the acceleration (as sensed by an accelerometer at the center of mass), $\boldsymbol{\tau} = [0\,0\,\tau]^T$ is the generated thrust, and $\mathbf{f}_{\text{ext}}$ is any external force applied to the aircraft. Note that since $\mathscr{B}$ is not an inertial frame, the gravity-induced force must not be included here.

We consider the following relations to convert a throttle command ρ to the exerted thrust τ.

$$\omega \propto \rho(t-\Delta t)\ , \quad \tau \propto c_\tau \omega^2 \tag{2}$$

where $\rho(t) \in [0,1]$ is the throttle command outputted by the autopilot, Δt is the time delay between a throttle command at time t and its actual application, $\omega \in [0,1]$ is interpreted as the normalized average setpoint angular velocities of the propellers and τ is the resulting thrust. The thrust coefficient c_τ links the angular velocity ω to the thrust τ.

C. Airspeed Definition

A fundamental concept when considering aerodynamic forces is the so-called "wind triangle", illustrated in Fig. 2. The wind vector $\mathbf{v}_w$ results from the difference between the ground velocity of the aircraft $\mathbf{v}_g$ (the velocity measured by a pose estimate) and the velocity relative to the air $\mathbf{v}_a$ (or airspeed), hence $\mathbf{v}_w = \mathbf{v}_g - \mathbf{v}_a$. Thus, the airspeed vector $\mathbf{v}_a = \begin{bmatrix} v_{ax} & v_{ay} & v_{az} \end{bmatrix}^T$ is responsible for the generation of drag forces.

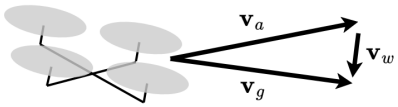

Fig. 2: Illustration of the wind triangle.

D. Lumped Drag Model

This section looks at different phenomena commonly identified as impacting the drag forces of a typical rotary-wing aircraft. Additionally, the complex underlying physics that motivates each model is summarized into lumped expressions that capture the general effect of each phenomenon. All the quantities mentioned in the following are assumed to be expressed in the body frame $\mathscr{B}$, unless otherwise specified.

1) Induced Drag: This drag is directly caused by the rotating blades of the MAV as they move through the air. The sector of the rotor traveling with high thrust (for the advancing blade) will generate more induced drag than the sector where the rotor generates less thrust (for the retreating blade) [23]. The induced drag is modeled as:

$$\mathbf{f}_i = -\tau \begin{bmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{v}_a \tag{3}$$

where d is the induced drag parameter in s/m and τ is the thrust. Note that some works consider this drag to be independent of the thrust [22].

2) Blade Flapping: The advancing blade has a higher absolute tip velocity and will generate more lift than the retreating blade. This generates an overall moment on the rotor disk in the direction of the apparent wind, which makes it tilt towards the wind direction, generating drag [23]. Additionally, some phase-lag effects can arise, and therefore

induce a tilt of the propeller not aligned with the wind direction. The blade flapping drag is modeled as:

$$\mathbf{f}_f = -\omega \begin{bmatrix} a_c & -a_s & 0 \\ a_s & a_c & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{v}_a \tag{4}$$

with the blade flapping coefficients a_c and a_s in $\mathrm{Ns}^2/\mathrm{m\,rad}$.

3) Axial Flow: When a propeller moves vertically, its thrust coefficient is affected. However, the impact on thrust can be highly non-linear. It is therefore more suitable to use the linearized formulation $c_\tau = c_{\tau 0} - \frac{c}{\omega r} v_{az}$ [20], which can be converted to a resulting drag force:

$$\mathbf{f}_a = -\omega \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c \end{bmatrix} \mathbf{v}_a \tag{5}$$

with the axial flow coefficient c in $\mathrm{Ns}^2/\mathrm{m\,rad}$.

4) Translational Lift: As a propeller moves through the air, the radial airspeed causes an increase in lift. This is known as translational lift [21]. The generated lift is expressed as:

$$\mathbf{f}_l = \tau\, l \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{v}_a \right\| \mathbf{u}_z \tag{6}$$

with l the translational lift coefficient in $\mathrm{s/m}$, and $\mathbf{u}_z$ the unit vector along the z-axis.

5) Parasitic Drag: This is the drag induced by the non-lifting surfaces of the quadrotor (e.g., the air frame, the motors, the embedded electronics, etc.) [22].

$$\mathbf{f}_p = -\begin{bmatrix} \mu_x & 0 & 0 \\ 0 & \mu_y & 0 \\ 0 & 0 & \mu_z \end{bmatrix} \mathbf{v}_a \odot |\mathbf{v}_a| \tag{7}$$

with the parasitic drag coefficients μ_x, μ_y and μ_z in $\mathrm{Ns}^2/\mathrm{m}^2$.

The final generic lumped drag model therefore includes eight coefficients and is given by:

$$\mathbf{f}_{\text{drag}}(\omega, \tau, \mathbf{v}_a \,|\, d, a_c, a_s, c, l, \mu_x, \mu_y, \mu_z) = \mathbf{f}_i + \mathbf{f}_f + \mathbf{f}_a + \mathbf{f}_l + \mathbf{f}_p \tag{8}$$

III. METHODS

This section describes how the coefficients of the lumped drag model are identified. The selected approach is to perform autonomous flights in wind-free environments, which yields the useful relation $\mathbf{v}_a = \mathbf{v}_g$, where $\mathbf{v}_g$ is obtained by deriving once with respect to time the pose provided by the state estimate of the aircraft. This lets us actively excite the dynamics along each axis of the aircraft within the limits of its flying capacity. Different trajectories are performed (see Section III-C below) to obtain the necessary data for the identification of the drag model parameters. Fig. 4 illustrates the resulting drag prediction for a wind-free autonomous flight.

A. Accelerometer Bias Estimation

The accelerometer bias is identified online by computing a continuously updated mean of the obtained accelerations. Since the computations are performed in the body frame $\mathscr{B}$, the gravity vector must be projected. The kth update of the bias $\bar{\mathbf{a}}$ with measured acceleration $\mathbf{a}$ is:

$$\bar{\mathbf{a}}[k] = ((\kappa - 1)\bar{\mathbf{a}}[k-1] + \mathbf{a} - {}^{\mathscr{B}}\mathbf{g})/\kappa, \text{ with } \kappa = \min(k, k_{\max}) \tag{9}$$

Furthermore, the value of κ can be limited to an upper bound $k_{\max}$ to allow the bias to evolve even after collecting a large amount of measurements. In this work, we use $k_{\max} = 1000$.

B. Throttle to Thrust Model

The thrust exerted by the MAV must be known according to (1); however, direct measurement of the applied thrust requires hardware adaptations and is therefore not always a suitable solution. Instead, we rely on the throttle command sent by the autopilot. First, the delay Δt between the throttle command and its application must be estimated. This delay can arise if the accelerometer data are filtered or can be caused by the ESCs or rotors' inertia. The throttle signal is correlated with the accelerometer data along the z-axis to identify the delay. Ideally, data from several flights with large vertical accelerations should be recorded, and the correlation between throttle commands and accelerations should be computed for each of them. Once the delay is known, the throttle commands ρ can be shifted in time to account for the estimated delay Δt, which yields the rotors' normalized rate ω, as defined in (2). Subsequently, the rate is mapped to thrust as follows:

$$\tau = \tau_0 + \tau_1 \omega + \tau_2 \omega^2 \tag{10}$$

where τ_0, τ_1 and τ_2 are positive constants to identify. In general, the relation between the rotor velocity and thrust is purely quadratic and one would expect to have $\tau_0 = \tau_1 = 0$. However, some autopilot settings can alter this relation, and therefore the more generic model in (10) is required.

C. Trajectories Generation

Trajectories must be appropriately defined to allow observation of the drag forces on the system along each axis.

For the vertical axis, setpoints are sent to the MAV along the vertical axis at fixed time intervals to excite its vertical dynamics. The x and y coordinates are kept constant.

To excite the horizontal dynamics of the MAV, we use piecewise minimum jerk trajectories, where each segment is described by a five-degree polynomial [27]. These can smoothly interpolate a series of arbitrary waypoints, and therefore generate arbitrarily complex smooth trajectories. This offers greater flexibility than classical parametric curves, such as the Lissajous curves. At each waypoint, the zeroth derivative is set at the waypoint location, while the second derivative is set to zero. For the first derivative, given a series of consecutive waypoints $\mathbf{w}_i$, $\mathbf{w}_j$ and $\mathbf{w}_k \in \mathbb{R}^3$, and a reference velocity v_w at the waypoints, we define the two unit vectors $\mathbf{e}_i$ and $\mathbf{e}_j$ along the directions $\mathbf{w}_j - \mathbf{w}_i$ and

$\mathbf{w}_k - \mathbf{w}_j$ respectively. The velocity vector at the waypoint $\mathbf{w}_j$ is calculated as $\mathbf{v}_j = \frac{1}{2}(\mathbf{e}_i + \mathbf{e}_j)v_w$. This process is repeated for each triplet of consecutive waypoints. Finally, the duration of each segment formed by a pair of consecutive waypoints $\mathbf{w}_i$ and $\mathbf{w}_j$ is set according to $||\mathbf{w}_j - \mathbf{w}_i||/v_s$, with a reference velocity v_s for the segments.

We use the sequence of waypoints illustrated in Fig. 3, which equally excites the x- and y-axes in both directions, with both lateral and diagonal velocities when flying with a fixed heading (in this case aligned with the x-axis).

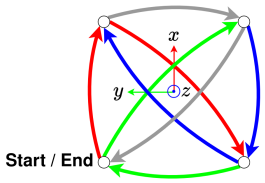

Fig. 3: Waypoints sequence used for the identification. The repeating pattern forming the trajectory is indicated by the sequence of colors (with the order: red, green, blue, gray).

Trajectories resulting from the sequence of waypoints described in Fig. 3 can be seen in Figs. 4 (real trajectory performed by the MAV) and 6 (computed trajectories).

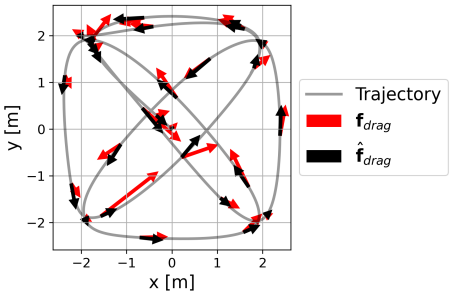

Fig. 4: MAV trajectory with measured drag forces $\mathbf{f}_{\text{drag}}$ and modeled drag forces $\hat{\mathbf{f}}_{\text{drag}}$ during a flight in no-wind conditions, reaching ground velocities of up to 2.5 m/s.

D. Identification with Gradient Descent

We perform the identification of the model parameters offline using a gradient-descent approach. Equations (1), (8) and (10) can be combined to define a loss function L, where each array is of size $(3, N)$ with N the number of samples:

$$L = \frac{1}{N}\sum_{n=0}^{N-1} \left\| m\mathbf{a}[:,n] - \boldsymbol{\tau}[:,n] - \mathbf{f}_{\text{drag}}[:,n] \right\| \qquad (11)$$

Here ':' represents the slice notation to indicate all values are considered along the specified axis. With this loss, a gradient is calculated for each coefficient defined in (8) and (10), using an autograd library, in our case PyTorch [28]. We use the Adam optimizer to update the coefficients at each iteration. Depending on the trajectory, gradient computations can be enabled or disabled for a subset of coefficients. Therefore, a single loss can be used to identify different sets of parameters. Furthermore, limits are set to the parameters to avoid unfeasible values and enforce constraints.

We decompose the identification problem into two phases. First, the parameters affecting the vertical axis are estimated, and second, the ones affecting the horizontal plane are identified. This two-step approach allows us to perform more suitable trajectories to identify the parameters. Indeed, the impact of the parameters can be difficult to distinguish when estimated all at once, further motivating this two-phase approach. More concretely, we execute purely vertical trajectories and estimate the delay Δt between throttle commands and perceived vertical accelerations. Then, the vertical parameters c, μ_z, τ_0, τ_1, and τ_2 are identified. Once known, these parameters are kept constant and the remaining horizontal parameters d, a_c, a_s, l, μ_x and μ_y are estimated by running trajectories based on the waypoints in Fig. 3, involving motions both in the horizontal and vertical planes.

For both phases, we performed multiple flights, each flight yielding an associated dataset of recorded onboard data. Furthermore, the estimation process is repeated five times for each flight and the identified parameters are each time initialized with random values uniformly distributed in the range $[0, 1]$, while the other non-estimated parameters remain unchanged. Repeating the identification with different initial values allows us to verify whether the gradient descent algorithm consistently converges towards the same solution. If this is not the case for certain parameters, they should be removed from the model. This constitutes our first criterion for removing redundant model parameters. The second criterion we employ is the correlation between identified parameters across flights, which means that the effect of certain parameters is compensated for by others. Finally, the third and last criterion used consists in verifying the compliance of a parameter with its physical meaning (e.g., obtaining strictly positive parasitic drag coefficients). This qualitative process enables us to eventually converge towards a minimal and physically anchored drag model.

IV. EXPERIMENTAL SETUP

This section introduces the MAV used in this work, as well as key characteristics of its sensor data. Finally, we describe the experiments carried out.

A. Micro Aerial Vehicle

To conduct real-world experiments, the Starling platform produced by ModalAI Inc., illustrated in Fig. 5, was used. It spans 190 mm diagonally and weighs slightly more than 300 g. It includes the companion computer VOXL, built around the Snapdragon 821 quad-core processor. Its base operating system is Linux Yocto Jethro with 3.18 kernel and runs ROS. The board contains two IMUs. The MAV is equipped with a flight core, joined to the main board, running PX4 and interfaced using MAVROS. State estimation is performed by monocular VIO, using a 45° downward looking global shutter wide-angle camera, running Qualcomm's proprietary VIO solution mvVISLAM.

The PX4 autopilot is limited to a velocity of 3 m/s and can accelerate up to 5 m/s^2 horizontally and 3 m/s^2 vertically.

Fig. 5: Starling MAV by ModalAI Inc.

B. Sensor Data

The data collected over MAVROS are listed in Table I. For brevity, the standard deviations (std) are reported for the norm of multidimensional measurements. The accelerometer standard deviation was measured under hover conditions, as the accelerometer noise is significantly affected by the motors. The attitude is provided by the VIO pose estimate, and its standard deviation was also measured while hovering.

TABLE I: Sensor Data Characteristics

Type	Unit	Rate Hz	Cutoff Freq. Hz	std.
Acceleration	m/s^2	50	5	1.378
Throttle	%	10	-	-
Attitude	rad	30	-	0.002
Position	m	30	-	0.023
Velocity	m/s	30	2	0.023

The standard deviations for position and velocity (obtained by differentiating the position) were determined by assessing the estimated trajectory generated by the VIO pipeline in relation to the ground truth trajectory acquired through a Motion Capture System (MCS) with a millimeter accuracy from Motion Analysis Inc. Prior to the comparison, the VIO estimated trajectory is aligned with the ground truth trajectory using a rigid transformation, minimizing the distance between both trajectories. We calculate the standard deviation of the error between the VIO estimated pose and the MCS ground truth for both the position and velocity.

C. Experiments

Following the type of trajectories described in Section III-C, three types of autonomous flights were carried out: vertical, horizontal, and combined.

The vertical experiments were composed of a sequence of vertically aligned targets sent at fixed time intervals of 1 and 1.5 seconds. We used three waypoints at a height of 1, 2, and 3 meters respectively in various sequences, which yields height changes between 1 and 2 meters in both directions. In horizontal experiments, we used the sequence of waypoints shown in Fig. 3 with fixed heading, with a displacement of ± 2 m along each axis. The reference velocity for the segments v_s is set at 2 m/s, and the velocity at the waypoints v_w is set at 1 m/s. The acceleration at the waypoints is set at 0 m/s^2. Finally, in the combined experiments, we add a height variation to the horizontal settings between each consecutive waypoint. The height was set to alternate between 1 and 3 m. This third configuration adds more diversity to the collected data and should provide greater robustness to the final identification results. The horizontal and combined configurations are shown in Fig. 6.

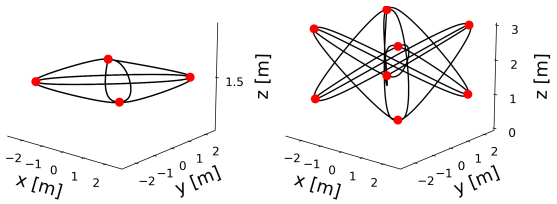

Fig. 6: Horizontal and combined trajectories derived from the waypoints sequence of Fig. 3.

V. RESULTS

This section covers the identification process carried out and presents the identified coefficients of the model.

A. Vertical Drag Model Identification

Six flights with pure vertical motions were performed. Using these data, the time delay between the throttle command and its effect on the vertical acceleration was estimated to be $\Delta t = 0.05 \pm 0.01$ s by computing the correlation between the throttle commands and the measured vertical accelerations.

The thrust model (10) could be identified, where the quadratic term converged to zero, in favor of the constant and linear terms. As the parasitic drag along the z-axis μ_z was highly correlated with the axial flow drag coefficient c (correlation of -0.99), we chose to drop μ_z. The identified vertical drag coefficients are summarized in Table II.

B. Correlation Between Drag Forces and Velocity

With the identified thrust model, we use (1) to calculate the external forces exerted on the MAV as $\mathbf{f}_{\text{ext}} = m(\mathbf{a} - \bar{\mathbf{a}}) - \boldsymbol{\tau}$. During free-flights, these forces should be correlated with the velocity of the aircraft, since they correspond to drag forces in this case. To verify this, we performed a combined trajectory, as shown in Fig. 6, which produced the Spearman correlation matrix in Fig. 7. The matrix clearly indicates that the MAV should be able to identify the aerodynamic forces exerted during its movements, as there is a clear correlation between its velocity and the perceived forces.

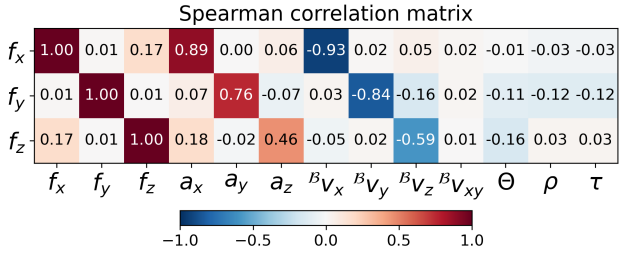

Fig. 7: Spearman correlation between observed external forces and sensor data. ${}^{\mathcal{B}}v_{xy}$ is the horizontal velocity norm.

C. Horizontal Drag Model Identification

Similarly to the vertical experiments, data were collected for eight flights. Four of the flights were performed with horizontal trajectories, while the other four included height variations (cf. Fig. 6). One dataset with height variations was

not included in the identification process and was used for validation, whose results are reported in Figs. 4 and 8.

After the first identification, the blade flapping coefficient a_c could not be consistently identified between runs and was fully correlated (-1) with the induced drag d. Thus, it was dropped, as well as a_c, which took a small value -0.006 ± 0.009 Ns^2/mrad. The identification was rerun and the parasitic drag coefficient μ_x showed a correlation of -0.91 with the induced drag d, while μ_y converged to a small negative value -0.008 ± 0.002 Ns^2/m^2. Both coefficients were removed. Rerunning the identification produced a translational lift coefficient l with a negative value of -0.002 ± 0.006 s/m, which is in contradiction with the physical meaning of the coefficient. It was also dropped. Therefore, the only valid drag coefficient remaining is the induced drag coefficient d.

D. Identified Drag Model

Table II summarizes the retained parameters, their identified values, and standard deviations. For the MAV considered, the propellers play an important role in generating drag, while the body of the aircraft does not have a significant impact. Hence, only the induced drag and axial flow play a significant role in drag generation.

TABLE II: Identified Parameters

Parameter	Value	std.	Unit
d	0.024	0.001	s/m
c	0.598	0.049	Ns/m
τ_0	0.923	0.029	N
τ_1	8.137	0.144	Ns/rad
Δt	0.05	0.01	s

The validation dataset (trajectory with height variations, see Fig. 6) was used to evaluate the drag forces due to the motion of the MAV itself as predicted by the drag model, versus the forces measured from the accelerometer and the applied thrust. We show the resulting forces in Fig. 8.

The drag model clearly follows the trend of the measured drag forces despite the noise present in the measured forces. This confirms that the identified drag captures the main characteristics of the true underlying drag. We performed a similar evaluation on the six vertical, four horizontal, and four combined trajectories executed for the identification, whose averaged results are reported in Table III.

TABLE III: Mean statistics across flights of force error $\mathbf{e} = \mathbf{f}_{\text{drag}} - \hat{\mathbf{f}}_{\text{drag}}$ in N

	$\lvert e_x \rvert$	$\lvert e_y \rvert$	$\lvert e_z \rvert$	$\lVert \mathbf{e} \rVert$
Mean	0.084	0.123	0.169	0.226
Max	0.660	0.980	1.366	1.822
RMSE	0.118	0.173	0.238	0.319

It is therefore possible to distinguish motion-induced drag forces from external forces, caused by wind or physical contact with an object. In this case, it is also easy to invert the drag model and recover the wind speed by solving (8) for $\mathbf{v}_a$ and then computing the wind speed as $\mathbf{v}_w = \mathbf{v}_g - \mathbf{v}_a$. Note that solving (8) may require one to solve a non-linear system of equations depending on which parameters of the model

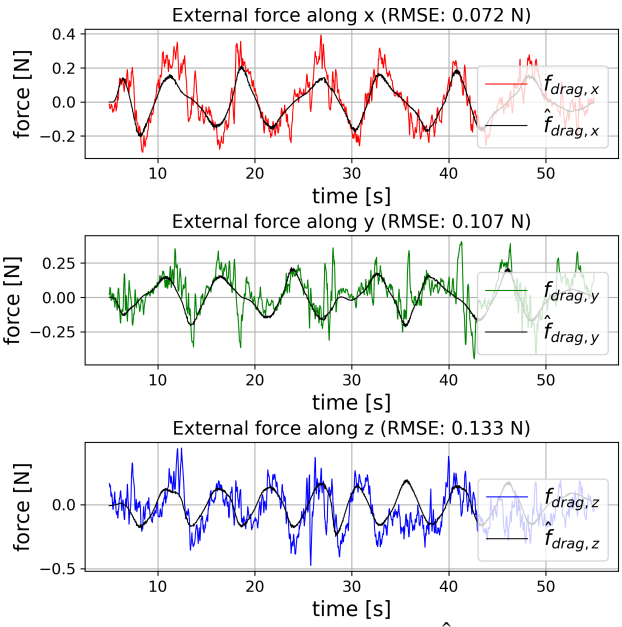

Fig. 8: Measured ($\mathbf{f}_{\text{drag}}$) vs. predicted ($\hat{\mathbf{f}}_{\text{drag}}$) drag forces during an autonomous flight with no-wind conditions.

are kept. This is not the case here, as only the induced drag and the axial flow drag are included and depend linearly on $\mathbf{v}_a$.

VI. CONCLUSION

In this work, we covered the main aerodynamic drag forces exerted on a rotary-wing aircraft and proposed a lumped formulation for each of them. This allowed us to perform an efficient gradient-based identification of the model parameters from free-flight onboard data in no-wind conditions without any prior assumption. The identification was carried out in two steps, first by identifying vertical and then horizontal drag parameters. The model complexity is minimized by iteratively removing redundant model parameters during the identification process. Although in this work we only worked with the Starling MAV, our approach is not restricted to a specific platform, as it solely relies on generally available onboard data a rotary-wing MAV can provide. Specific trajectories were proposed to collect data suitable for the identification of a MAV's drag parameters, using vertically aligned waypoints and smooth minimum jerk trajectories. External forces are measured by direct observation of accelerations and throttle commands, making the approach lightweight and easily implementable in various systems. Furthermore, this allows us to create the baseline to capture external wind- or contact-induced disturbances (see the video at the link provided in the first page footnote, where we show the detection of perturbations manually applied to the MAV). As suggested by the results of this work, the main sources of drag for a rotary-wing MAV are its propellers, namely, the propellers' induced drag and axial flow drag. In the future, we will conduct systematic experiments in a wind tunnel to quantitatively verify the measurement of wind perturbation forces.

References

[1] R. T. Palomaki, N. T. Rose, M. v. d. Bossche, T. J. Sherman, and S. F. J. D. Wekker, "Wind Estimation in the Lower Atmosphere Using Multirotor Aircraft," *Journal of Atmospheric and Oceanic Technology*, vol. 34, no. 5, pp. 1183–1191, May 2017, Publisher: American Meteorological Society Section: Journal of Atmospheric and Oceanic Technology. DOI: 10.1175/JTECH-D-16-0177.1.

[2] R. Felismina, M. Silva, A. Mateus, and C. Malça, "Study on the aerodynamic behavior of a UAV with an applied seeder for agricultural practices," in *AIP Conference Proceedings*, vol. 1836, Rome, Italy, 2017, p. 020049. DOI: 10.1063/1.4981989.

[3] G. M. Dering, S. Micklethwaite, S. T. Thiele, S. A. Vollgger, and A. R. Cruden, "Review of drones, photogrammetry and emerging sensor technology for the study of dykes: Best practises and future potential," *Journal of Volcanology and Geothermal Research*, vol. 373, pp. 148–166, Mar. 2019. DOI: 10.1016/j.jvolgeores.2019.01.018.

[4] M. Stokkeland, K. Klausen, and T. A. Johansen, "Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection," in *International Conference on Unmanned Aircraft Systems*, Jun. 2015, pp. 998–1007. DOI: 10.1109/ICUAS.2015.7152389.

[5] T. Sankey, J. Donager, J. McVay, and J. B. Sankey, "UAV lidar and hyperspectral fusion for forest monitoring in the southwestern USA," *Remote Sensing of Environment*, vol. 195, pp. 30–43, Jun. 2017. DOI: 10.1016/j.rse.2017.04.007.

[6] S. Jung, S. Song, S. Kim, *et al.*, "Toward Autonomous Bridge Inspection: A framework and experimental results," in *16th International Conference on Ubiquitous Robots*, ISSN: 2325-033X, Jun. 2019, pp. 208–211. DOI: 10.1109/URAI.2019.8768677.

[7] D. Thakur, G. Loianno, W. Liu, and V. Kumar, "Nuclear Environments Inspection with Micro Aerial Vehicles: Algorithms and Experiments," in vol. 11, arXiv:1903.06111 [cs], 2020, pp. 191–200. DOI: 10.1007/978-3-030-33950-0_17.

[8] J. Scherer, S. Yahyanejad, S. Hayat, *et al.*, "An Autonomous Multi-UAV System for Search and Rescue," in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, Florence Italy: ACM, May 2015, pp. 33–38. DOI: 10.1145/2750675.2750683.

[9] S. Prudden, A. Fisher, M. Marino, A. Mohamed, S. Watkins, and G. Wild, "Measuring wind with Small Unmanned Aircraft Systems," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 176, pp. 197–210, May 2018. DOI: 10.1016/j.jweia.2018.03.029.

[10] A. Tagliabue, A. Paris, S. Kim, R. Kubicek, S. Bergbreiter, and J. P. How, "Touch the Wind: Simultaneous Airflow, Drag and Interaction Sensing on a Multirotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, ISSN: 2153-0866, Oct. 2020, pp. 1645–1652. DOI: 10.1109/IROS45743.2020.9341797.

[11] M. Marino, A. Fisher, R. Clothier, S. Watkins, S. Prudden, and C. S. Leung, "An Evaluation of Multi-Rotor Unmanned Aircraft as Flying Wind Sensors," *International Journal of Micro Air Vehicles*, vol. 7, no. 3, pp. 285–299, Sep. 2015, Publisher: SAGE Publications Ltd STM. DOI: 10.1260/1756-8293.7.3.285.

[12] P. P. Neumann and M. Bartholmai, "Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit," *Sensors and Actuators A: Physical*, vol. 235, pp. 300–310, Nov. 2015. DOI: 10.1016/j.sna.2015.09.036.

[13] T. Tomić, K. Schmid, P. Lutz, A. Mathers, and S. Haddadin, "The flying anemometer: Unified estimation of wind velocity from aerodynamic power and wrenches," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, ISSN: 2153-0866, Oct. 2016, pp. 1637–1644. DOI: 10.1109/IROS.2016.7759264.

[14] J.-Y. Wang, B. Luo, M. Zeng, and Q.-H. Meng, "A Wind Estimation Method with an Unmanned Rotorcraft for Environmental Monitoring Tasks," *Sensors*, vol. 18, no. 12, p. 4504, Dec. 2018, Number: 12 Publisher: Multidisciplinary Digital Publishing Institute. DOI: 10.3390/s18124504.

[15] J. González-Rocha, C. A. Woolsey, C. Sultan, and S. F. J. De Wekker, "Sensing Wind from Quadrotor Motion," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 4, pp. 836–852, Apr. 2019. DOI: 10.2514/1.G003542.

[16] G. Perozzi, D. Efimov, J.-M. Biannic, and L. Planckaert, "Using a quadrotor as wind sensor: Time-varying parameter estimation algorithms," *International Journal of Control*, vol. 95, no. 1, pp. 126–137, Jan. 2022, Publisher: Taylor & Francis. DOI: 10.1080/00207179.2020.1780324.

[17] K. Meier, R. Hann, J. Skaloud, and A. Garreau, "Wind Estimation with Multirotor UAVs," *Atmosphere*, vol. 13, no. 4, p. 551, Mar. 2022. DOI: 10.3390/atmos13040551.

[18] S. Waslander and C. Wang, "Wind Disturbance Estimation and Rejection for Quadrotor Position Control," in *AIAA Infotech@Aerospace Conference*, Seattle, Washington: American Institute of Aeronautics and Astronautics, Apr. 2009. DOI: 10.2514/6.2009-1983.

[19] Y. Qu, K. Wang, and X. Wu, "Wind Estimation with UAVs Using Improved Adaptive Kalman Filter," in *Chinese Control And Decision Conference*, ISSN: 1948-9447, Jun. 2019, pp. 3660–3665. DOI: 10.1109/CCDC.2019.8832809.

[20] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a quad-rotor robot," in *Proceedings of the Australasian Conference on Robotics and Automation*, B. MacDonald, Ed., Australia: Australian Robotics and Automation Association, 2006, pp. 1–10.

[21] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *IEEE International Conference on Robotics and Automation*, ISSN: 1050-4729, May 2009, pp. 3277–3282. DOI: 10.1109/ROBOT.2009.5152561.

[22] M. Bangura and R. Mahony, "Nonlinear Dynamic Modeling for High Performance Control of a Quadrotor," *Australian Robotics and Automation Association*, 2012, Victoria University of Wellington, New Zealand.

[23] R. Mahony, V. Kumar, and P. Corke, "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, Sep. 2012. DOI: 10.1109/MRA.2012.2206474.

[24] H. Chen and H. Bai, "Incorporating thrust models for quadcopter wind estimation," *IFAC-PapersOnLine*, 2nd Modeling, Estimation and Control Conference, vol. 55, no. 37, pp. 19–24, Jan. 2022. DOI: 10.1016/j.ifacol.2022.11.155.

[25] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, "Data-Driven MPC for Quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, Apr. 2021. DOI: 10.1109/LRA.2021.3061307.

[26] G. Cioffi, L. Bauersfeld, and D. Scaramuzza, *HDVIO: Improving Localization and Disturbance Estimation with Hybrid Dynamics VIO*, arXiv:2306.11429 [cs], Jun. 2023. DOI: 10.48550/arXiv.2306.11429.

[27] C. Richter, A. Bry, and N. Roy, "Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor

Environments," in *Robotics Research*, M. Inaba and P. Corke, Eds., vol. 114, Series Title: Springer Tracts in Advanced Robotics, Cham: Springer International Publishing, 2016, pp. 649–666. DOI: 10.1007/978-3-319-28872-7_37.

[28] A. Paszke, S. Gross, F. Massa, *et al.*, *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, arXiv:1912.01703 [cs, stat], Dec. 2019. DOI: 10.48550/arXiv.1912.01703.