Capturing Accurate Quadrotor Dynamics

Alexander Bouman
Mechanical Engineering Department
Carnegie Mellon University
Pittsburgh, Pennsylvania United States
abouman@andrew.cmu.edu

***Abstract*—As quadrotors become more popular and users begin to demand more agility, accurate dynamics models for use in controlling and simulating quadrotors will become increasingly important. When flying a quadrotor at high speeds, the drag of the quadrotor's body and rotors as well as changes in wing loading begin to dominate the forces acting on the aircraft. The goal of this work is to describe a simple and effective method for modeling the forces on a quadrotor with a fast, differentiable model suitable for use in controlling quadrotors at high speed.**

***Index Terms*—Quadrotors, Dynamics, Modeling**

Supplementary Material

All code and data used in this project is available at https://github.com/aabouman/AeroFitting.

I. Introduction

Recently years quadrotors have become an increasingly popular platform of choice for unmanned aerial vehicles. One of the most appealing attributes of quadrotors is their relatively simple design. Unlike most other rotorcraft, quadrotors do not have any mechanical linkages. With its bare-bone design, quadrotors cannot rely on a mechanical system for passive stability, instead, they are dependent on the on-board computation of motor input informed by a model of the aircraft's dynamics. Because of this, accurately modeling the quadrotor's dynamics is critical to their control.

At high speeds, quadrotor's become hard to control due to the complexities of modeling rotors and aerodynamic forces on the quadrotor body. This becomes even more challenging when downwash from rotors interact and when the aircraft flys through its downwash.

Traditionally, quadrotors are approximated around hover conditions with thrust and drag modeled as linear functions of motor speed [4]. These models are smooth and computationally inexpensive but fail to capture the effect of a quadrotors velocity in quadrotor lift, lift induced drag, and body drag. Borrowing from the developed theory of helicopter rotor dynamics, some researchers have modeled quadrotors using Blade-Element-Momentum (BEM) theory. BEM is able to capture the changing lift and drag behavior of a rotor with airspeed, however, it is difficult to implement and computationally more expensive. In this work we model the net forces acting on the body as a general quadratic function of quadrotor body and motor velocities and demonstrate this model's performance.

A. Quadrotor Dynamics

The quadrotor's dynamics are given by the dynamics of a 6-DOF rigid body with control forces, F_B, and moments, τ_B applied on the body. Equation 1 describes the quadrotor kinematics and dynamics using quaternions to describe the quadrotor's attitude, giving the state-space a total of 13 dimensions.

$$\dot{x}_B = \begin{pmatrix} \dot{p} \\ \dot{Q} \\ \dot{v}_B \\ \dot{\omega}_B \end{pmatrix} = \begin{pmatrix} Qv_B \\ \frac{1}{2}L(Q)H\omega_B \\ \frac{1}{m}F_B - \omega_B \times v_B + Q^T g \\ J^{-1}\left(\tau_B - \omega_B \times J\omega_B\right) \end{pmatrix} \tag{1}$$

Here p gives the position of the quadrotor wrt to a world frame, Q is a unit quaternion representing the attitude of the body wrt the world, and v_B and ω_B denote the linear and rotational velocity of the quadrotor measured in the body frame. The function $L(\cdot)$ denotes left multiplication of a quaternion and $g = \begin{pmatrix} 0 & 0 & -9.81 m/s^2 \end{pmatrix}^T$ is the acceleration of gravity written in the world frame.

The focus of this work is modeling the force and moment applied at the body frame, F_B and τ_B respectively. From the design and geometry of a quadrotor we can decompose this body wrench into individual rotor components as shown in equations 2, 3.

$$F_B = F_{res} + \sum_{l \in \text{Motors}} F_l \tag{2}$$

$$\tau_B = \tau_{res} + \sum_{l \in \text{Motors}} \tau_l + r_l \times F_l \tag{3}$$

The forces and torques F_{res} and τ_{res} are the un-modeled forces applied to the aircraft. These are mostly dominated by bluff body aerodynamic forces. The vector, r_l describes the location of the rotor l relative to the quadrotor's center of mass.

A common model for F_l and τ_l is a quadratic function of the rotational speed of motor l shown in 4 and 5.

$$F_l = \begin{pmatrix} 0 & 0 & C_L \, \Omega_l^2 \end{pmatrix}^T \tag{4}$$

$$\tau_B = \begin{pmatrix} 0 & 0 & C_D \, \Omega_l^2 \end{pmatrix}^T \tag{5}$$

These models are derived from the classic lift and drag models in aerodynamics modeling the forces on an airfoil as a function of air velocity over the blade. Due to their simplicity, they are easy to implement and empirically fit C_L and C_D to data

collected from a motor test stand. While these models work well in slow flight and hover conditions, they fail to accurately capture the forces on the quad at higher speeds. In particular, these models fail to account for the asymmetry of incident air velocity on a rotor blade at high speeds. When a quadrotor is performing aggressive maneuvers the velocity of each rotor's advancing blade (the side of the rotor swinging into the air stream) produces more lift and drag than the rotor's retreating blade.

B. Blade Element Momentum Theory

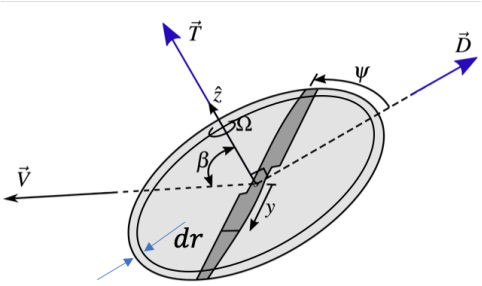

Fig. 1: Rankine rotor disk, with thrust, drag and orientation of rotor labeled [3]

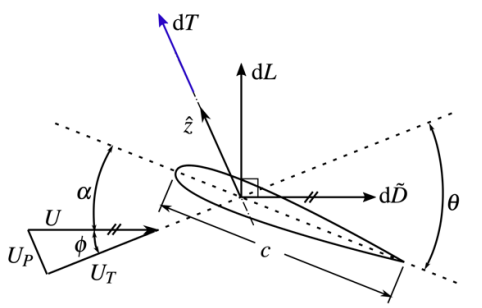

Fig. 2: Forces applied to infinitesimal section of rotor blade. [3]

A more complex method that can account for the difference in relative airspeed across a rotor is Blade-Element-Momentum (BEM) Theory [3] [2] [4] [1]. BEM isolates each rotor and decomposes the forces on an infinitesimal piece of the rotor over an infinitesimal arc the rotor sweeps out over a full revolution. In this way BEM can use traditional lift drag functions such as those shown in 9 and 10. The net thrust (force perpendicular to the rotor disk) and drag (force in rotor disk plane aligned with velocity of disk – $\vec{V}$ in figure 1) are then modeled in equations 6, 7, and 8. Where the constant ρ is the density of air, and b is the number of blades on the rotor.

$$T = \frac{b\rho}{4\pi}\int_0^R\int_0^{2\pi}\left(dL\cos\phi + dD\sin\phi\right)\,d\Psi\,dr \tag{6}$$

$$H = \frac{b\rho}{4\pi}\int_0^R\int_0^{2\pi}\left(-dL\sin\phi + dD\cos\phi\right)\sin\Psi\,d\Psi\,dr \tag{7}$$

$$D = \frac{b\rho}{4\pi}\int_0^R\int_0^{2\pi}\left(-dL\sin\phi + dD\cos\phi\right)r\,d\Psi\,dr \tag{8}$$

Examining the infinitesimal functions in 9 and 10, we see that these are still the same quadratic with velocity functions of lift and drag with fitted constants C_L and C_D, and characteristic length, in this case chord length, $c(r)$.

$$dL(r,\Psi) = c(r)\,C_l(\alpha(r,\Psi))\,U(r,\Psi)^2 \tag{9}$$

$$dD(r,\Psi) = c(r)\,C_d(\alpha(r,\Psi))\,U(r,\Psi)^2 \tag{10}$$

The modeling of C_L and C_D is no longer as trivial as in the previous simple quadratic models specified in 4 and 5. Now the force coefficients must be modeled as a function blade angle of attack α. This requires either a test stand which is capable of varying and measuring the angle of attack of a rotor blade or approximating the $C_l(\alpha(r,\Psi))$ and $C_D(\alpha(r,\Psi))$ as a simple closed-form function of α and the coefficients of lift and drag at steady hover flight, such as done in [3].

II. Proposed Model

While BEM models can greatly outperform the simple hover quadratic models, they pose significant difficulty in implementation and ultimately still rely on accurate empirical data collected from custom test stands to measure the coefficients of lift and drag. Fundamentally the thrust and drag equations of 6 8 are fitted quadratic models, with the extra complexity of decomposing the net force and torque vectors into parallel and perpendicular components to the rotor disk and quadrotor heading.

Embracing this gray-box approach to modeling quadrotor forces we can then model the net force and torques on the quadrotor as a second-order polynomial, dependent on all of the systems velocities.

A. Modelling Forces

Revisiting the original quadrotor dynamics from 1, we can compute the net forces on our quad as F_B.

$$\dot{v}_B = \frac{1}{m}F_B - \omega_B \times v_B - Q^T g \tag{11}$$

$$F_B = m\left(\dot{v}_B + \omega_B \times v_B + Q^T g\right) \tag{12}$$

We then model the net force on the quadrotor as a quadratic function of the quadrotor's velocities. This is expressed as $\hat{F}_B(x)$ where $x \in \mathbb{R}^{10}$ containing the quadrotor linear and angular velocities as well as the rotational velocities of each rotor.

$$x = \begin{pmatrix} v_B^T & \omega_B^T & \Omega^T \end{pmatrix}^T \tag{13}$$

$$\hat{F}_B(x) : \mathbb{R}^{10} \to \mathbb{R}^3 \tag{14}$$

The function $\hat{F}_B(x)$ then is a quadratic with form given in 15. This form is used to capture the classic lift and drag functions. Equation 15 then expresses this function using Einstein tensor notation. It is important to note that A is a vector, B is a matrix, and C is a 3rd order tensor.

$$\left(\hat{F}_B(x)\right)_i = A_i + \sum_{j=1}^{\#x} B_{i,j}x_j + \sum_{j=1}^{\#x}\sum_{k=1}^{\#x} C_{i,j,k}x_j x_k \tag{15}$$

We can then rewrite this problem as a linear problem by transforming the input vector by f as shown in 16, where the last term is the Kronecker product of x with itself. This gives a vector of all possible cross-coupled second-order terms of x.

$$z = f(x) = \begin{pmatrix} \mathbf{1} \\ x \\ x \otimes x \end{pmatrix} \tag{16}$$

Corresponding to the transformation in input to $z = f(x)$, the coefficients of the polynomial $\hat{F}_B(x)$ can then be reformatted as the matrix D by unraveling the tensor C into a matrix.

$$D \in \mathbb{R}^{3\times\left(1+\#x+(\#x)^2\right)} \tag{17}$$

$$D = \begin{pmatrix} \underbrace{A}_{3\times 1} & \underbrace{B}_{3\times \#x} & \underbrace{\begin{pmatrix} C_{i,j,1} & \dots & C_{i,j,\#x} \end{pmatrix}}_{3\times(\#x\cdot\#x)} \end{pmatrix} \tag{18}$$

Finally this can be expressed as the linear problem:

$$\hat{F}_B(x) = D\, f(x) = D\, z \Longrightarrow \hat{F}_B(x)^T = D^T z^T \tag{19}$$

B. Modelling Torques

We use a similar method to model the torques as the forces with the addition of motor accelerations to the input of the function $\hat{\tau}_B(x)$. We now can compute the total moment on the quadrotor using 21.

$$\dot{\omega}_B = J^{-1}\left(\tau_B - \omega_B \times J\omega_B\right) \tag{20}$$

$$\tau_B = J\dot{\omega}_B + \omega_B \times J\omega_B \tag{21}$$

The torque about a quadrotor's yaw is dominated by the changing angular momentum of the aircraft's rotors. To capture this the modeled torque $\hat{\tau}_B(x)$ must include the derivative of motor velocities as an input.

$$x = \begin{pmatrix} v_B^T & \omega_B^T & \Omega^T & \dot{\Omega}^T \end{pmatrix}^T \tag{22}$$

$$\hat{\tau}_B(x) : \mathbb{R}^{14} \to \mathbb{R}^3 \tag{23}$$

Again modelling the function $\hat{\tau}_B(x)$ as a quadratic we use the same transform $f(x)$ to linearize the function. Using the same method as described in 18 and **??**. The resulting model is then:

$$\hat{\tau}_B(x)^T = E^T f(x)^T \tag{24}$$

Where

C. Fitting The Model

Using a dataset published by [2], we selected three flights spanning different flight regimes of a quadrotor to use for a training dataset. The first flight, whose position trajectory is given in 3, is an aggressive linear oscillation of the quadrotor. This flight in particular forces the quadrotor to rapidly switch motor speeds as well as drives the quadrotor through its propwash. Flight 2 consists of a wobbly circle with more aggressive moments applied, and flight 3 is a 3D circle forcing the quadrotor through changes in the world Z axis. These flights are displayed in figures 4 and 6 respectively.

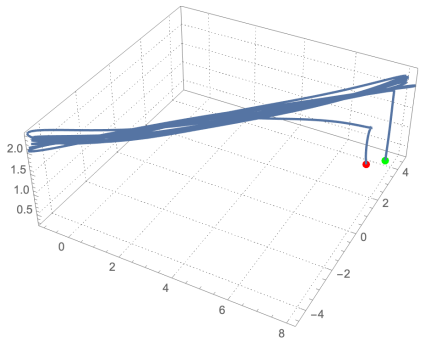

Fig. 3: Quadrotor position training flight 1. Red point designates the take off location and green landing location.

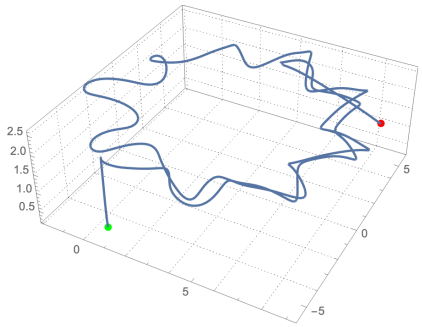

Fig. 4: Quadrotor position training flight 2. Red point designates the take off location and green landing location.

Using the dataset from [2] we extract the data corresponding to the input of functions $\hat{F}_B(x)$ and $\hat{\tau}_B(x)$ specified in 13 and 22. Transforming these inputs using 16, we compute a matrix of inputs z_i. Similarly using equations 12 and 21 we compute a matrix of outputs y_i. The matrices D and E are then the solution of the linear least squares problem.

III. Model Performance

A. Training

Empirically we can evaluate the quality of the model using the coefficient of determination, the ratio of the variance

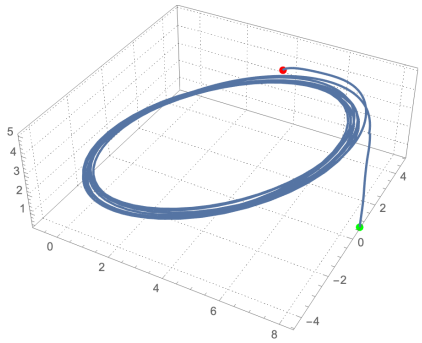

Fig. 5: Quadrotor position training flight 3. Red point designates the take off location and green landing location.

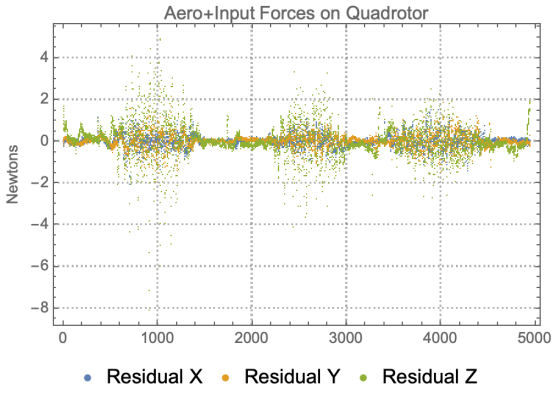

Fig. 6: Force model training residuals.

explained by the model output to total variance in the ground truth.

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{25}$$

$$R^2 = 1 - \frac{\sum_i \left\| y_i - \hat{F}_B(x_i) \right\|}{\sum_i \|y_i - \bar{y}\|} \tag{26}$$

Similarly for the goodness of fit of the torque model:

$$R^2 = 1 - \frac{\sum_i \|y_i - \hat{\tau}_B(x_i)\|}{\sum_i \|y_i - \bar{y}\|} \tag{27}$$

The result of this goodness of fit test on the force and torque models is outlined in table I. From the goodness of fit of the force model we see that $\hat{F}_B(x_i)$ accurately models the training data. The torque model does not perform nearly as well, with a

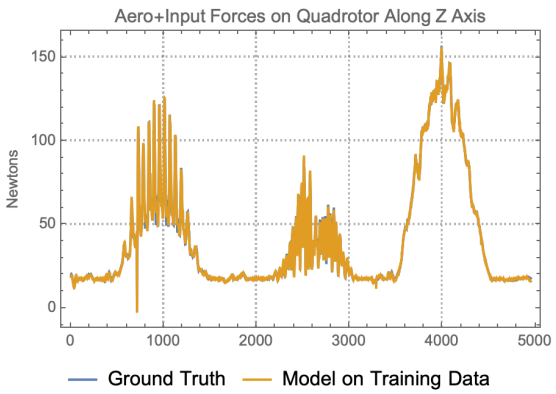

Fig. 7: Forces along quadrotor Z axis during training flights.

training $R^2 = 0.2859$. The major reason for this is most likely due to a lack of an accurate inertia matrix. The dataset from [2] did not specify the mass and inertia properties of the quadrotor. Unlike the force, the inertia tensor is not linearly applied to the entire expression. This is clear from the expression 21. The code and quadrotor model associated with the dataset [2] could not be found, thus the inertia tensor that was used to compute $y_i = J\dot{\omega}_B + \omega_B \times J\omega_B$ was taken from a paper published by the same authors 2 months earlier [5], in the hopes that the same quadrotor was used for both papers. It is suspected this had a large impact on the inability to fit the quadrotor's moments to a quadratic function.

In total, the training flights consisted of 120 seconds of flight time sampled at 400Hz. This data was then down-sampled to 200Hz before learning model parameters.

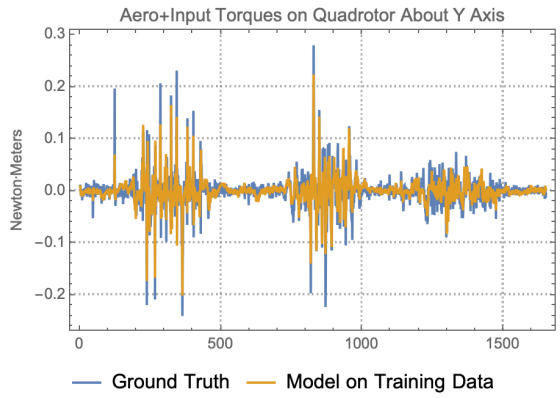

Fig. 8: Torques about quadrotor Z axis during training flights.

B. Testing

TABLE I: Model Fit Quality

Flight	Coefficient of Determination	
	Force	*Moment*
Training	0.977039	0.285944
Testing 1	0.972797	0.0937357
Testing 2	0.961393	0.194875

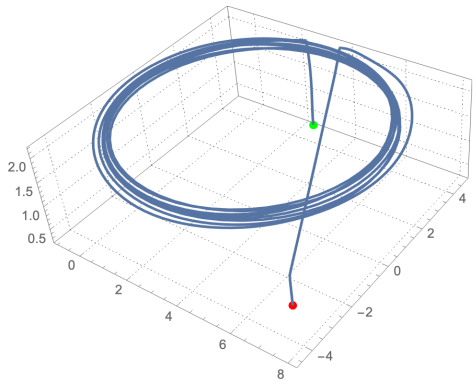

Fig. 9: Quadrotor position during test flight 1. Red point designates the take off location and green landing location.

Following construction and computation of model parameters D and E the models shown in equations 24 and 19. These models were tested against two data sets that did not appear in the training data. These test flights' position trajectory is shown in figures 9 and 10. The forces predicted and experienced on these flights are displayed in the plots 11 and 13. The force model performs well and has a coefficient

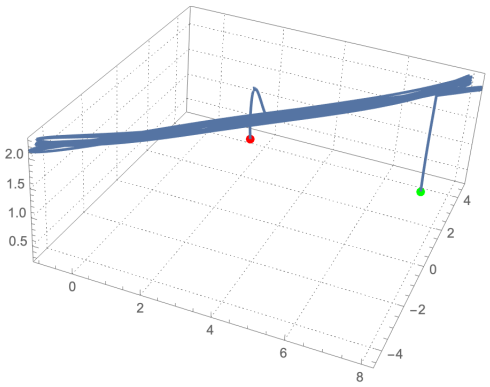

Fig. 10: Quadrotor position during test flight 2. Red point designates the take off location and green landing location.

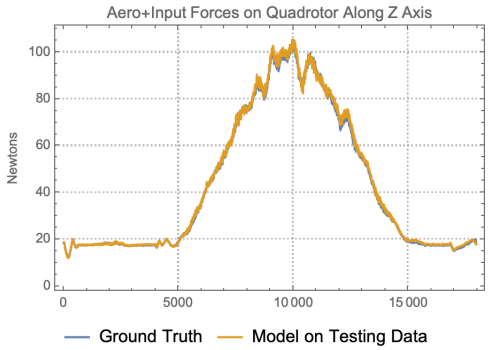

Fig. 11: Forces along quadrotor Z axis during testing flight 1.

of determination near that of the training data, suggesting the model has not been overfitted to the training dataset and generalizes well.

In addition to the least-squares fit for the quadratic models of force and torque, we are interested in using different methods to fit the quadratic model to the data. In particular, a random sample consensus (RANSAC) based algorithm would be better at rejecting noise and outliers in the data and is an iterative method, enabling easier learning of the model parameters with new data.

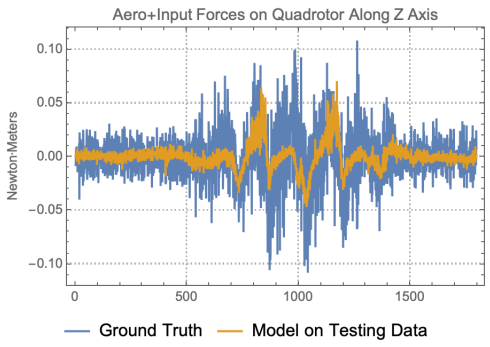

Fig. 12: Torques about quadrotor Z axis during testing flight 1.

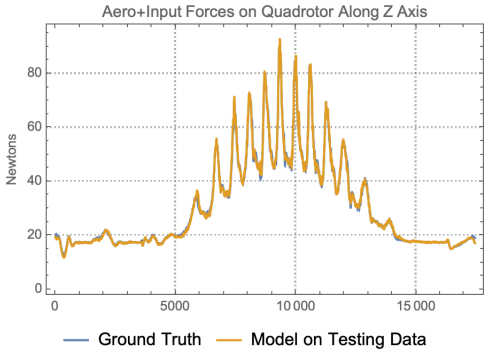

Fig. 13: Forces along quadrotor Z axis during testing flight 2.

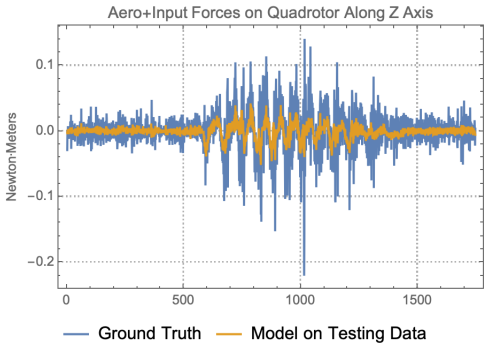

Fig. 14: Torques about quadrotor Z axis during testing flight 2.

Additionally, we are interested in independently modeling each motor's force such that rather than having to model and learn the entire function 3. Instead, we learn just the pure moment, τ_l each motor rotor pair l contributes to the total moment.

IV. Conclusion

From this work, we have seen the efficacy of a simple quadratic model of the forces on a quadrotor as a function of its body and motor velocity. This work is particularly interesting as it allows for rapid bootstrapping in force modeling on a quadrotor. Starting from a simple linear hover model, flying a quadrotor for two minutes resulted in enough data to greatly improve the accuracy of the quadrotor's dynamics model.

This simple poly-fit method has a few major advantages over more complex BEM-based methods. As a single matrix multiplication, the function is incredibly low cost. Additionally, as a second-order polynomial, it has continuous and easily computed first and second derivatives, a useful property with many applications in controlling the quadrotor.

References

[1] Moses Bangura, Marco Melega, Roberto Naldi, and Robert E. Mahony. Aerodynamics of rotor blades for quadrotors. *arXiv: Fluid Dynamics*, 2016.

[2] Leonard Bauersfeld, Elia Kaufmann, Philipp Foehn, Sihao Sun, and Davide Scaramuzza. Neurobem: Hybrid aerodynamic quadrotor model. *RSS: Robotics, Science, and Systems*, 2021.

[3] Rajan Gill and Raffaello D'Andrea. Propeller thrust and drag in forward flight. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 73–79, 2017.
[4] Gabriel M. Hoffmann, Haomiao Huang, Steven L. Waslander, and Claire J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. 2012.
[5] Guillem Torrente, Elia Kaufmann, Philipp Foehn, and Davide Scaramuzza. Data-driven mpc for quadrotors. *IEEE Robotics and Automation Letters*, 2021.