

ONLINE AERODYNAMIC IDENTIFICATION WITH KALMAN FILTERING FOR DISTINCT PROPELLER CHARACTERISTICS

SPENCER FOLK [SFOLK@SEAS], ALEXANDER SPINOS [SPINOS@SEAS]

ABSTRACT. In this work, we develop a computationally light estimator for a quadrotor with unknown and possibly distinct propeller aerodynamics using measurements from an onboard gyroscope. An Extended Kalman Filter is used to estimate several model parameters related to the aerodynamics of each individual propeller. Better estimation of the actuator dynamics can lead to improved trajectory tracking, and relaxing the assumption that these parameters are known and constant is especially useful in cases where a quadrotor is cobbled together from a variety of recycled motors and propellers, or if a motor or propeller becomes damaged during an operation. In this paper, we show that these parameters are observable and present simulated experiments using a modified version of the MEAM 620 quadcopter simulator. In simulation, the filter rapidly converges to accurate parameter values, and in some cases greatly improves tracking performance. In the future, we will further investigate the impact of filtering-based estimators on system stability when used online in the control loop. Additionally, experiments will be carried out to demonstrate the effectiveness of the estimator on real hardware.

1. INTRODUCTION

Small unmanned aerial vehicles (sUAVs), referred to informally as drones, have continued to gain popularity among both enterprise and consumer populations, in large part due to continued reductions in cost and improvements in sensing, actuation, and battery technologies. Besides being a popular research platform for robotic design, low level control, and motion planning [1], drones have demonstrated their value in a wide variety of tasks ranging from entertaining stadium light shows [2] and precision agriculture [3] to providing critical surveillance and intelligence information in modern conflicts [4].

Despite the diversity of morphologies that drones exhibit, the vast majority of drones today are all actuated in a similar manner: small Direct Current electric motors drive air propellers at variable rates that, when orchestrated by a control system, produce a desired net thrust and torque on the body. Inherent in this method of actuation is a required knowledge of the mechanics describing propeller force and torque generation, since without a proper model of the actuator dynamics a control system would never produce a desired wrench on the robot. To this end, rather than using unwieldy expressions found in rotorcraft literature [5], roboticists have historically used simplifying assumptions, like near-hover operating conditions with no wind and an identical set of undamaged propellers, to reduce the actuator dynamics down to quadratic models that relate a propeller's thrust and torque to the square of its rotation rate with two fitting parameters [6]. These parameters, frequently referred to as the *thrust* and *drag* coefficients, are fitted once for a given propeller using static thrust stand tests. It is at this point where most roboticists will stop thinking about the actuator dynamics of a drone.

For applications that don't have high trajectory tracking performance standards or otherwise operate in ideal conditions, the simplified quadratic models for thrust and torque are—and have historically been—sufficient. However, as drones take on more autonomy in more constrained [7] and chaotic [8] arenas, accurate actuator models play a pivotal role in mission success. Decades worth of research in rotorcraft has yielded plenty of more accurate propeller models of varying fidelity, from lower fidelity models like Blade Element Momentum Theory (BEMT) [5] to full-fledged numerical solutions to the Navier-Stokes equation [9]. Unlike the roboticists' two-parameter models, these methods can model the effects from high winds (or equivalently high speed flight), rotor-rotor interaction, and rotor-surface interaction. The trade-off for these more expressive models is of course that they become too complex and cumbersome for applications in control or motion planning. Further, one glaring issue is that even more complex models still assume a fixed blade geometry, which may not be the case for a propeller subject to low manufacturing tolerances, degraded from prior use, or even damaged mid-flight. **For high performance applications that require low tracking errors, drones require expressive and adaptive models of their actuator dynamics that are amenable to traditional methods for control and motion planning.**

1.1. Contributions. The primary contribution of this project is a computationally tractable Extended Kalman Filter that estimates individual propeller thrust and drag coefficients using only measurements from an on-board gyroscope.

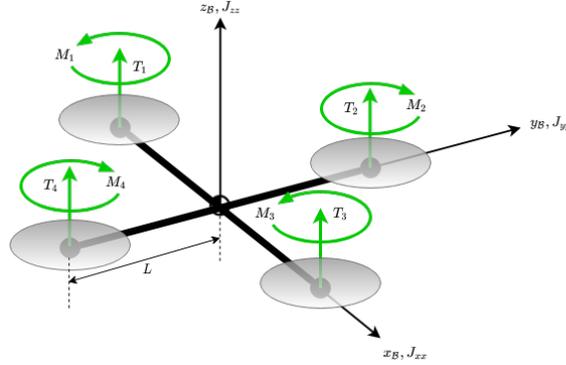


FIGURE 1. A standard “plus” configuration quadcopter with arm length L .

Our proposed solution to the need for expressive yet lightweight actuator dynamic models preserves the controller- and planning-friendly quadratic models for propeller thrust and torque but enhances the models by implicitly relaxing the assumption that the thrust and drag coefficients are constant and indistinct from one propeller to the next. And unlike related works, our filter can likely run on even highly constrained processors and converges to accurate estimates of propeller coefficients within 200 samples, potentially making it feasible for a controller to use these estimates in the loop for small agile quadcopters.

2. RELATED WORK

When concerning system identification of the actuator dynamics for a multi-rotor vehicle, the standard approach is to secure a motor and propeller to a thrust stand and perform a series of static tests¹. This requires very little domain knowledge to carry out, but the trade-off is that the fitted parameters are only accurate for the specific propeller and wind conditions tested. Gill et al. [10] took a brute force approach and collected data over 19 propellers subject to various wind conditions using a wind tunnel. The result was a lumped parameter model with coefficients for each propeller that are functions of relative airflow. Another brute force approach is found in Ahmad et al. [9], where the thrust and drag coefficients are computed using Computational Fluid Dynamics (CFD). Both approaches highlight the laborious nature of system identification if one wants to capture all the different flow conditions a propeller may experience.

Thanks to increasing compute resources, model-based online parameter identification of UAV actuator dynamics has become a very active field of research. Munguia et al. [11] use an Extended Kalman Filter to successfully identify a single thrust coefficient for all actuators on a quadcopter both in simulation and hardware using IMU measurements. Works from Svacha et al. [12],[13] identify both a motor torque coefficient and thrust coefficient based on IMU data using an Unscented Kalman Filter, but similarly these values are assumed to be the same among all four propellers and are not compared against any ground truth measures. Recently Böhm et al. [14] presented a Bayesian filter that recursively estimates not only navigation states, sensor biases, and inertial parameters, but also thrust and drag coefficients for individual propellers. Results from [14] are done in simulation and the filter requires both IMU and position measurements. This paper most resembles the anticipated contributions of our work and was only discovered after most of the project was completed. Importantly, all of the above papers incorporate a nonlinear observability analysis to varying extents to ensure filter convergence.

3. BACKGROUND

This problem considers the standard quadcopter with four propellers equidistant from the center of mass by length L , as seen in Figure 1. The body frame \mathcal{B} is aligned with the principal axes of the quadcopter ensuring that the inertia tensor is diagonal with elements $\{J_{xx}, J_{yy}, J_{zz}\}$.

Each propeller produces a thrust, T_i , and moment, M_i , which are modeled as

$$T_i = k_f^{(i)} \eta_i^2 \quad M_i = k_m^{(i)} \eta_i^2 \quad (1)$$

¹https://kumar-robotics-github-documentation.readthedocs.io/en/master/references/hardware/thrust_stand/thrust_stand.html

This model is a slight but meaningful modification to convention, in that we assume that each propeller has a unique thrust (k_f) and drag (k_m) coefficient and there is no requirement that these parameters are constant in time.

4. APPROACH

Our approach uses a compact model representation informed by a nonlinear observability analysis (Section 4.1) to estimate distinct thrust and drag coefficients using an Extended Kalman Filter. The process and measurement models are derived in Section 4.2, and the filter is evaluated in a simulation environment described in Section 4.3 using trajectories found in Section 4.4.

4.1. Observability Analysis and Variable Selection. We used a nonlinear observability analysis to determine which state variables and measurements are required to estimate the aerodynamic coefficients [15]. This technique is described in detail and applied to IMU estimation in prior work, see [12] and [14] for a more detailed description. The analysis requires that the process model is written in control affine form, $\dot{\mathbf{x}} = \mathbf{f}_0 + \sum_{i=1}^4 \mathbf{f}_i(\mathbf{x})\mathbf{u}^{(i)}$. Lie derivatives of the measurement model \mathbf{h} with respect to the dynamics are concatenated into an observability matrix defined in Equation 2.

$$\mathcal{O} = [(\nabla\mathbf{h})^\top, (\nabla\mathcal{L}_{\mathbf{f}_0}\mathbf{h})^\top, (\nabla\mathcal{L}_{\mathbf{f}_1}\mathbf{h})^\top, (\nabla\mathcal{L}_{\mathbf{f}_2}\mathbf{h})^\top, (\nabla\mathcal{L}_{\mathbf{f}_3}\mathbf{h})^\top, (\nabla\mathcal{L}_{\mathbf{f}_4}\mathbf{h})^\top]^\top \quad (2)$$

If $\text{rank}(\mathcal{O}) = n$ where $n = \dim(\mathbf{x})$ then the system is considered *locally weakly observable*. In other words, there exists a trajectory such that each state $\mathbf{x}^{(i)}$ is uniquely distinguishable from all other states $\mathbf{x}^{(j)}$, $j \neq i$

The observability matrix \mathcal{O} was computed with Mathematica [16] using a modified version of a notebook made available by Svacha et al. ² We tested several different combinations of states (position, orientation, velocity, aerodynamic coefficients, inertial parameters, IMU biases) and measurements (gyroscope, accelerometer) in order to find the most compact state representation such that $k_f^{(i)}$ and $k_m^{(i)}$ were observable. We found that a simple system composed of only the body angular rates, aerodynamic coefficients, and gyro measurements was observable. This analysis was insufficient to show that inertial parameters could also be estimated, so in this work we assume the inertial parameters are known.

4.2. Modeling. The state and control spaces are shown in Equation 3:

$$\mathbf{x} = [\boldsymbol{\omega}_B^\top, \mathbf{k}_f^\top, \mathbf{k}_m^\top, \mathbf{b}_g^\top]^\top \in \mathbb{R}^{14}, \quad \mathbf{u} = [\eta_1^2, \eta_2^2, \eta_3^2, \eta_4^2]^\top \quad (3)$$

where $\boldsymbol{\omega}_B \in \mathbb{R}^3$ is the quadrotor angular velocity in the body frame, $\mathbf{k}_f \in \mathbb{R}^4$ are the motor-propeller thrust coefficients, $\mathbf{k}_m \in \mathbb{R}^4$ are the motor-propeller moment coefficients, and $\mathbf{b}_g \in \mathbb{R}^3$ are the gyro biases. The control space \mathbf{u} is the vector of squared motor speeds, where η_i is the i th motor speed. Most electronic speed controllers (ESCs) used on quadcopters can provide the current motor speed, so it is reasonable to assume we have access to the true motor speed.

4.2.1. Process Model. The process model primarily consists of the body rate dynamics written in the quadcopter's body coordinate frame:

$$\dot{\omega}_x = \frac{1}{J_{xx}} \left[L(k_f^{(2)}u^{(2)} - k_f^{(4)}u^{(4)}) - (J_{zz} - J_{yy})\omega_y\omega_z \right] \quad (4)$$

$$\dot{\omega}_y = \frac{1}{J_{yy}} \left[L(k_f^{(1)}u^{(1)} - k_f^{(3)}u^{(3)}) + (J_{zz} - J_{xx})\omega_x\omega_z \right] \quad (5)$$

$$\dot{\omega}_z = \frac{1}{J_{zz}} \left[k_m^{(1)}u^{(1)} - k_m^{(2)}u^{(2)} + k_m^{(3)}u^{(3)} - k_m^{(4)}u^{(4)} - (J_{yy} - J_{xx})\omega_x\omega_y \right] \quad (6)$$

where $J_{(\cdot)}$ indicates the moment of inertia about the respective body axes and L is the distance between each propeller and the center of mass. The aerodynamic coefficients and gyroscope biases are difficult to model explicitly. Rather than attempting to capture these complexities, the process model assumes these states are constant.

$$\dot{\mathbf{k}}_f = 0 \quad \dot{\mathbf{k}}_m = 0 \quad \dot{\mathbf{b}}_g = 0 \quad (7)$$

By concatenating Equations 4-7 we arrive at a process model in the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. For the filter, these dynamics are discretized using Forward Euler and a timestep of $dt = 0.002$ seconds.

²https://github.com/jsvacha/observability_analysis

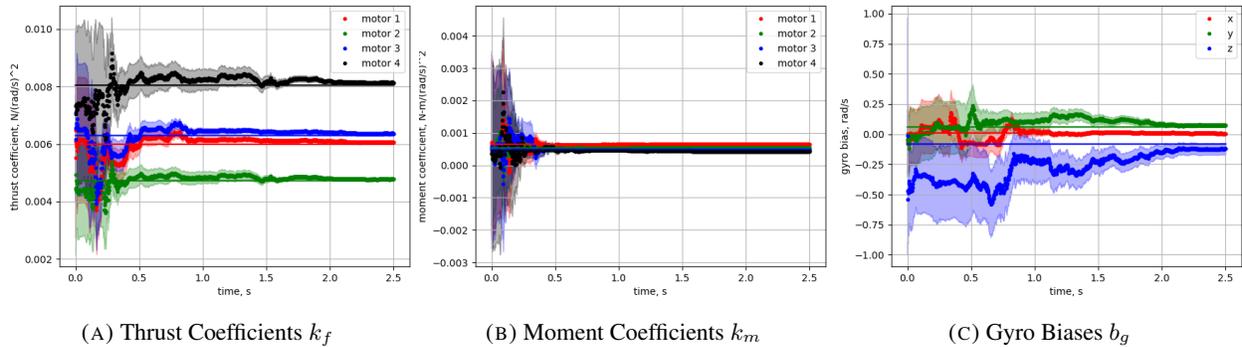


FIGURE 2. Example filter estimates during the first 2.5 seconds of the Lissajous trajectory. The shaded region shows one standard deviation of the estimated distribution.

4.2.2. *Measurement Model.* The nonlinear observability analysis provided the conclusion that only measurements from the gyroscope were necessary for the system $\{\mathbf{x}, \mathbf{u}\}$ to be observable. Thus, the measurement model is simple.

$$\mathbf{y} = \boldsymbol{\omega}_B + \mathbf{b}_g \quad (8)$$

Conveniently, the measurement model is already linear in the state, and so the measurement model can be written in the form $\mathbf{y} = \mathbf{h}(\mathbf{x}) = \mathbf{C}\mathbf{x}$.

4.3. **Simulation Environment.** Simulations were done using a modified version of the MEAM 620 Simulator. The simulator came with a basic model of a quadcopter and a nonlinear geometric position controller [17]. We modified the simulator to specify individual propeller parameters for each of the four actuators, and we added simulated IMU measurements with specified bias and noise parameters. Most importantly, we added the capability for a filter that receives measurements from the IMU. A flag can be set to switch the controller feedback from ground truth values or estimates from the filter. The simulator, filter, and controller loops are fixed at a rate of 500 Hz.

4.4. **Trajectory Design.** We tested the filter and controller on several trajectories of varying motion complexity. This is because it is important that the filter visits a variety of different states to avoid conditions that are unobservable, particularly around hover. Ideally, the trajectory should excite all axes of the gyroscope, especially yaw (z_B). The simplest trajectory tested is a *step* input whereby the quadcopter is commanded to translate in only the x -axis. The quadcopter does not yaw in this trajectory. Another simple but effective trajectory is a *circle* path with a back-and-forth desired yaw.

Consistent with [13] and [14], we also implemented a reparameterized compound *Lissajous* trajectory.

$$\begin{bmatrix} x_{\mathcal{W}}(t) \\ y_{\mathcal{W}}(t) \\ z_{\mathcal{W}}(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} A_x(1 - \cos(2\pi n_x t/T)) \\ A_y \sin(2\pi n_y t/T) \\ A_z \sin(2\pi n_z t/T) \\ A_\psi \sin(2\pi n_\psi t/T) \end{bmatrix} \quad (9)$$

Each flat output is specified using trigonometric functions with different amplitudes $A_{(\cdot)}$ and frequencies using the number of cycles $n_{(\cdot)}$ in the full trajectory period T . To obtain variety in the signals produced by the gyroscope, we superimpose a large amplitude, low frequency Lissajous curve with a small amplitude, high frequency Lissajous curve to form the compound Lissajous trajectory. Each of the three trajectories is evaluated for 10 seconds.

5. EXPERIMENTAL RESULTS

The results show that the filter rapidly converges to the correct values with proper tuning. Figure 2 shows the first few seconds of the filter performance in one of the test cases. The moment coefficients are often recovered in a fraction of a second, while gyro biases are recovered after a few seconds. Detailed results are shown in Table 1.

The results cover the combination of the three trajectories (Lissajous, circle, and step) and three controller settings. The *Oracle* controller is given perfect knowledge of the position, orientation, velocity, and the propeller coefficients, providing a baseline measure of performance. The *Incorrect* controller is given incorrect coefficient estimates that are

		Tracking Error				Convergence Time		% stable
		x (m)	y (m)	z (m)	yaw ($^\circ$)	k_f & k_m (s)	b_g (s)	
Lissajous	Oracle	0.064	0.063	0.005	1.535	0.278	1.955	100
	Incorrect	0.158	0.161	0.076	11.569	0.180	1.968	90
	Filter	0.064	0.062	0.012	12.020	0.288	2.038	80
Circle	Oracle	0.047	0.045	0.001	0.600	0.083	3.057	100
	Incorrect	0.215	0.152	0.075	17.383	0.223	2.611	100
	Filter	0.056	0.054	0.013	28.442	0.172	2.426	100
Step	Oracle	0.230	0.000	0.001	0.000	0.019	9.539	100
	Incorrect	0.325	0.121	0.067	16.168	0.044	9.395	100
	Filter	0.308	0.088	0.159	26.371	0.051	8.232	100

TABLE 1. Simulation results. The table entries are the average values over 10 trajectories with randomly selected constant parameters and initial belief. Trajectories where the control became unstable were omitted from the average. Tracking error is root-mean-square-error (RMSE).

sampled from a Gaussian distribution around the true values. The *Filter* controller starts at the same incorrect values, but updates those values online using the filter estimates at a rate of 500 Hz.

In all nine cases, we simulate 10 flights with randomly sampled true values of the propeller coefficients and gyro biases. In each case, we evaluate the tracking error as well as the filter convergence time. We define the filter convergence time as the point at which all future estimates are within 0.1 standard deviation of the true value. This is split into the time for *all* propeller coefficients to converge and the time for *all* gyro biases to converge.

From the table, we can see that the convergence time for the propeller coefficients is fast regardless of the trajectory, within 200 observations from the IMU. The gyro biases often fail to converge during the step trajectory, because this trajectory does not excite yaw. It is also clear that tracking performance is negatively impacted by having the incorrect coefficients; *Incorrect* is significantly worse than *Oracle*. For the majority of cases, *Filter* almost completely recovers the XYZ-tracking performance. However, the yaw tracking is not improved when the controller uses the filter estimate.

6. DISCUSSION

In this report, we present an Extended Kalman Filter that estimates distinct thrust and drag coefficients for each propeller on a quadcopter. The filter is lightweight; there are only 14 filter states and only gyroscope measurements are necessary. This means this estimation scheme could likely run at a high frequency on most computationally-constrained platforms, which likely cannot be said for the highly related work by Böhm et al. [14] that has 68 filter states.

The results show that the filter is able to converge to within 0.1 standard deviations of the true propeller coefficients within 200 samples which is comparable to if not better than related works [13, 14]. Our filter’s sample efficiency—in part due to proper tuning but also due to its simple design—revealed a question regarding how a controller might perform when operating under the filter’s belief on the actuator dynamics. Our results suggest that for sufficiently exciting trajectories it is possible to nearly recover the average tracking performance of a controller that has perfect knowledge of the propeller coefficients. This conclusion could pave the way towards UAVs that can adapt to changing wind conditions or even mid-flight propeller failure without necessitating the design and validation of new control laws. The coupling between a nonlinear observer such as our EKF and a nonlinear controller remains to be carefully studied in this context.

One glaring result is the yaw tracking performance, which suffers greatly even with small inaccuracies in propeller drag coefficients. Using the filter’s estimate did not seem to lower average yaw tracking error; in fact, it seemed to make yaw tracking performance worse in all three trajectories evaluated. One possible explanation is that the controller’s yaw-axis gains require better tuning but this seems unlikely because the *Oracle* controller has excellent yaw tracking. Another more likely explanation reveals a hurdle with trying to combine nonlinear observers and controllers. The dynamics of our EKF are difficult to predict, and wide variations in the filter estimate of $k_f^{(i)}$ and $k_m^{(i)}$ could lead to very poor transient controller performance that is never fully recovered by the end of the trajectory.

In the future, we hope to investigate the stability of the controller and observer in the case where the actuator dynamics are unknown. We are also interested in validating the computational feasibility of our lightweight observer compared to other works on constrained hardware.

REFERENCES

- [1] M. W. Mueller, S. J. Lee, and R. D'Andrea, "Design and control of drones," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, p. null, 2022. [Online]. Available: <https://doi.org/10.1146/annurev-control-042920-012045>
- [2] I. Singh. (2021, August) Tokyo olympics closing ceremony: special effects, secret drone light show. [Online]. Available: <https://dronedj.com/2021/08/08/olympics-closing-ceremony-drone-light-show/>
- [3] DJI. (2020, November) Dji's latest agras t20 drone makes agricultural spraying easier, smarter, and safer. [Online]. Available: <https://ag.dji.com/newsroom/dji-ag-news-en-agras-t20>
- [4] D. Hambling. (2022, April) Small quadcopters rule the battlefield in ukraine — which makes their chinese manufacturers very unhappy. [Online]. Available: <https://www.forbes.com/sites/davidhambling/2022/04/29/small-quadcopters-rule-the-battlefield-in-ukraine---which-makes-their-chinese-manufacturers-very-unhappy/?sh=39b9a888768>
- [5] G. J. Leishman, *Principles of helicopter aerodynamics*. Cambridge university press, 2006.
- [6] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, p. 22, 2011.
- [7] E. Ackerman. (2021, April) Exyn brings level 4 autonomy to drones. [Online]. Available: <https://spectrum.ieee.org/exyn-brings-level-4-autonomy-to-drones>
- [8] S. Weintraub. (2022, May) Drone tornado footage from kansas shows unbelievable devastation. [Online]. Available: <https://dronedj.com/2022/05/01/drone-tornado-footage-kansas/>
- [9] F. Ahmad, P. Kumar, R. Dobriyal, and P. P. Patil, "Estimation of the thrust coefficient of a quadcopter propeller using computational fluid dynamics," *IOP Conference Series: Materials Science and Engineering*, vol. 1116, no. 1, p. 012095, apr 2021. [Online]. Available: <https://doi.org/10.1088/1757-899x/1116/1/012095>
- [10] R. Gill and R. D'Andrea, "Computationally efficient force and moment models for propellers in uav forward flight applications," *Drones*, vol. 3, no. 4, p. 77, 2019.
- [11] R. Munguía, S. Urzua, and A. Grau, "EKF-based parameter identification of multi-rotor unmanned aerial vehicles models," *Sensors*, vol. 19, no. 19, p. 4174, 2019.
- [12] J. Svacha, G. Loianno, and V. Kumar, "Inertial yaw-independent velocity and attitude estimation for high-speed quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1109–1116, 2019.
- [13] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, "Imu-based inertia estimation for a quadrotor using newton-euler dynamics," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3861–3867, 2020.
- [14] C. Böhm, M. Scheiber, and S. Weiss, "Filter-based online system-parameter estimation for multicopter uavs," *Proceedings of Robotics: Science and Systems XVII. Robotics: Science and Systems Foundation*, 2021.
- [15] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on automatic control*, vol. 22, no. 5, pp. 728–740, 1977.
- [16] W. R. Inc., "Mathematica, Version 13.0.0," champaign, IL, 2021. [Online]. Available: <https://www.wolfram.com/mathematica>
- [17] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 5420–5425.