Sliding Mode Control for Translational Trajectory Following for a Quadrotor Vehicle

William Selby

Abstract— **The quadrotor vehicle is an extremely versatile platform that utilizes the force differences in each of its four motors for rotational and translational positioning. This research focuses on the translational control of an Ascdening Technologies Humingbird quadrotor. In order to control the highly nonlinear dynamics of the quadrotor, a sliding mode controller is developed. This method of control is known to be robust to modeling errors. Utilizing this controller, simulation results are presented as well as experimental results which demonstrate the performance capabilities of this controller.**

I. INTRODUCTION

Advances in nonlinear control theory as well as hardware and software improvements have made it possible to precisely control highly nonlinear and underactuated systems such as the quadrotor vehicle. This platform is small and can be used for a variety of applications including exploring an unknown environment, providing sensor coverage, and aerial surveillance. Their low weight allows them to be equipped with various sensors and deployed into the environment.

Specifically, this research focuses on the translational control of a quadrotor vehicle for trajectory following. The high level goal is to develop an autonomous system for visual surveillance. While image segmentation, target tracking and target identification are related issues, this work assumes that there is a higher level controller creating a desired trajectory to place the quadrotor above the intended target.

Regardless of the application, at the low level, the quadrotor needs to be able to follow a specified trajectory accurately while also being robust to noise and disturbances from the environment. Sliding mode control was chosen for its ability to stabilize the platform in the face of unknown modeling errors. The control scheme operates by forcing an error vector toward a hypersurface in the state space [1]. Once on the hypersurface, the vehicles dynamics are defined by the dynamics of the surface. These dynamics converge the error vector towards zero.

A. Related Work

Extensive modeling of the quadrotor as well as methods of model parameter estimation have been investigated such as in [2], [3], [4], and [5].

Previous work has shown successful implementation and performance of PID controllers for translational motion shown in [6] and [3]. [7] compared using LQ control techniques to PID controllers. [8] and [2] created controllers for the quadrotor platform using a sliding mode and extracted the controllers using backstepping. [5] also used sliding modes to control the quadrotor, but computed desired roll and pitch angles as the controller output instead of calculating motor speeds.

II. NONLINEAR MODEL OF THE QUADROTOR

A depiction of the quadrotor used for the model development is shown in Figure 1 depicting the Euler angles of roll (ϕ), pitch (θ), and yaw (ψ) and the cartesian coordinate frame. The motors operate in pairs with motors 1 and 3 operating together and motors 2 and 4 operating together. By varying the speed of a motor, it is possible to manipulate the generalized lift force. Changing the relative speeds of motors 1 and 3 controls the pitch angle, and consequently creates motion in the x axis. Similarly, varying the speeds of motors 2 and 4 creates a roll angle which creates translational motion in the y axis. Altitude is controlled by the sum of the forces of all the motors. Lastly, a

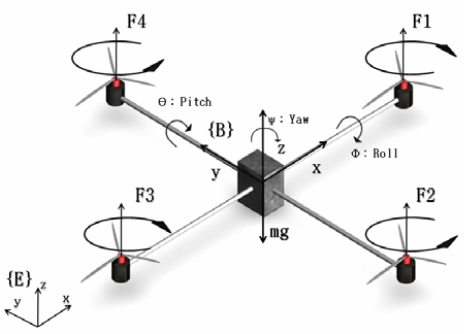

Fig. 1. Quadrotor Model [2]

yaw moment is created from the difference in the counter-torques between each pair of motors.

This model relies on several assumptions [2]. First, the quadrotor structure is rigid and symmetrical with a center of mass aligned with the center of the body frame of the vehicle. The force created by the motors is assumed to be proportional to the motor speed and the propellers are considered to be rigid. An estimated model is compared with ground-truth accelerations to validate the model as well as the assumptions.

The dynamic equations can be derived by Newtons laws of motion. First, divide the coordinate system into an earth reference frame (E) and a body reference frame (B) as shown in Figure 1.

$$\begin{aligned} m\ddot{x} &= F_f + F_t + F_g \\ J\ddot{\theta} &= \Gamma_f + \Gamma_a \end{aligned} \tag{1}$$

Here, m is the mass of the quadrotor and J is a constant, symmetric, and positive definite rotational inertia matrix with respect to the body frame of the quadrotor defined by Equation 2.

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{2}$$

The rotational transformation matrix between the earth reference frame and the body reference frame utilizing the Euler angles is defined by Equation 3 were C and S represent the cosine and sine trigonometric functions.

$$R_E^B = \begin{bmatrix} C_\theta C_\psi & C_\psi S_\theta S_\phi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\psi S_\phi \\ C_\theta S_\psi & S_\psi S_\theta S_\phi - C_\phi C_\psi & C_\phi S_\theta S_\psi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{3}$$

F_f is the resultant force generated by the four motors defined in equation 4 where the force of each motor, F_i, is defined in equation 5 as the product of the lift coefficient K_p and the square of the motor's angular velocity ω.

$$F_f = R_E^B \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} F_i \end{bmatrix} \tag{4}$$

$$F_i = K_p \omega^2 \tag{5}$$

Additionally, F_t is the resultant of the drag forces along each translational axis defined in equation 6.

$$F_t = \begin{bmatrix} -K_{ftx} & 0 & 0 \\ 0 & -K_{fty} & 0 \\ 0 & 0 & -K_{ftz} \end{bmatrix} \dot{x} \tag{6}$$

K_{ftx}, K_{fty}, and K_{ftz} are the drag coefficients. F_g is the gravity force vector defined in equation 7.

$$F_g = \begin{bmatrix} 0 & 0 & -mg \end{bmatrix}^T \tag{7}$$

Γ_t is the moment developed by the quadrotor in the body frame expressed in equation 8.

$$\Gamma_t = \begin{bmatrix} d(F_3 - F_1) \\ d(F_4 - F_2) \\ K_d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \tag{8}$$

Finally, Γ_a is the resultant torques of aerodynamic friction shown in equation 9 where d is the distance from the motor to the center of mass of the quadrotor.

$$\Gamma_a = \begin{bmatrix} -K_{fax} & 0 & 0 \\ 0 & -K_{fay} & 0 \\ 0 & 0 & -K_{faz} \end{bmatrix} \dot{\theta} \quad (9)$$

The inputs to the traditional model derived above are the angular velocities of the four motors as shown in equation 10

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} K_p & K_p & K_p & K_p \\ -K_p & 0 & K_p & 0 \\ 0 & -K_p & 0 & K_p \\ K_d & -K_d & K_d & -K_d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (10)$$

These equations result in the following equations of motion described in equation 11.

$$\begin{aligned} \ddot{x} &= \frac{(C_\phi S_\theta C_\psi + S_\phi S_\psi)u_1 - K_{ftx}\dot{x}}{m} \\ \ddot{y} &= \frac{(C_\phi S_\theta C_\psi - S_\phi C_\psi)u_1 - K_{fty}\dot{y}}{m} \\ \ddot{z} &= \frac{(C_\phi C_\theta)u_1 - K_{ftz}\dot{z}}{m} - g \\ \ddot{\phi} &= \frac{du_2 - K_{fax}\dot{\phi}}{I_x} \\ \ddot{\theta} &= \frac{du_3 - K_{fax}\dot{\theta}}{I_y} \\ \ddot{\psi} &= \frac{u_4 - K_{faz}\dot{\psi}}{I_z} \end{aligned} \quad (11)$$

A. *Model Adaption*

Utilizing the Ascending Technologies platform has several distinct advantages. Primarily, one can use the onboard autopilot system to control the attitude of the quadrotor. This allows the user to focus on translational control. The autopilot inputs are a desired roll and pitch angle, desired yaw rate, and a generalized thrust command. The remainder of this research replaces the rotational dynamics in equation 11 with the dynamics of the autopilot. For translational motion, the inputs become a desired roll, pitch and yaw value to create an attitude that translates the quadrotor to the desired position. The autopilot and motor controllers compute the rotor velocities resulting in the inputs $\{u_1 ... u_4\}$

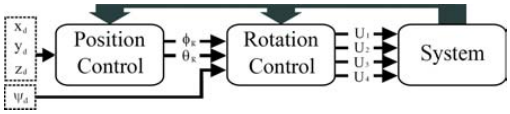

Fig. 2. Quadrotor Control Scheme [8]

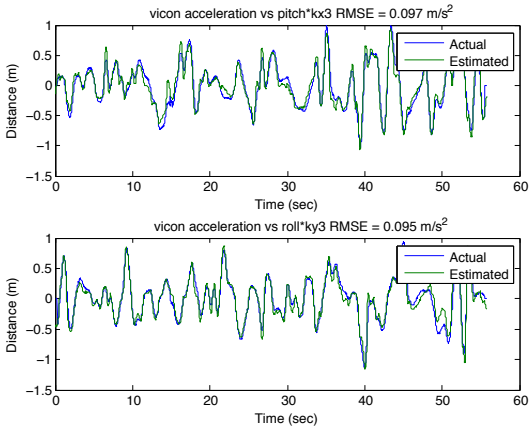

Fig. 3. Quadrotor Model Comparison

as defined above. This resulting control scheme is summarized in Figure 2.

One of the main advantages of robust control is the ability of the controller to overcome modeling errors. Due to the highly nonlinear system of equations of motion for the quadrotor, it is necessary to reduce the complexity of the model based on the assumptions given above. To verify the validity of those assumptions, an estimated model was computed from groundtruth data to determine the accuracy and bounds of the parameters in the estimated model. Figure 3 compares the results of an estimated translational model of the system to ground-truth data captured by a Vicon motion capture system with sub-millimeter accuracy.

The estimated model is shown in equation 12 with the calculated paramters.

$$\begin{aligned} \ddot{\hat{x}} &= 10.84(C_\phi S_\theta C_\psi + S_\phi S_\psi) - 0.37\dot{\hat{x}} + 0.16 \\ \ddot{\hat{y}} &= -10.49(C_\phi S_\theta C_\psi - S_\phi C_\psi) - 0.18\dot{\hat{y}} - 0.28 \end{aligned} \quad (12)$$

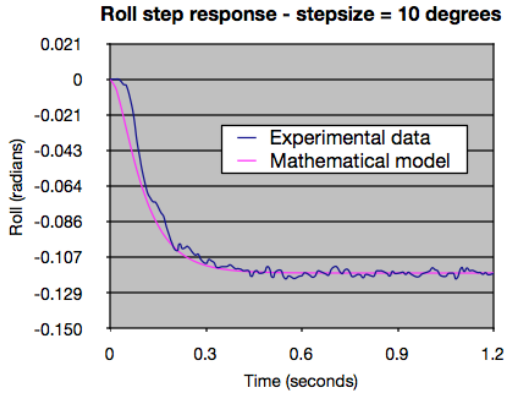

Fig. 4. Quadrotor Autopilot Roll Characterization

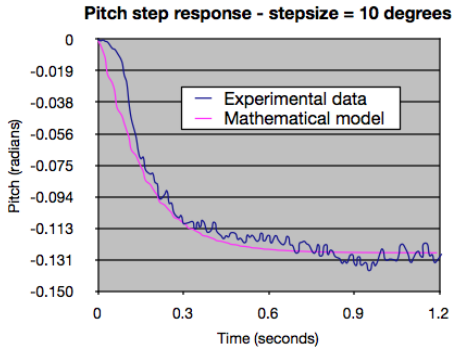

Fig. 5. Quadrotor Autopilot Pitch Characterization

The effectiveness of the control scheme also depends on the accuracy of the onboard autopilot system. Step responses were performed on the quadrotor with roll and pitch inputs and the resulting attitude was recorded with the Vicon system. The autopilot dynamics were estimated as equation 13. These equations were used in the controller simulations in place of the attitude equations derived in equation 11. These equations reflected the actual plant and was more accurate than the derived model equations. The step responses were recorded and analyzed with assistance from Daniel Soltero.

$$\begin{aligned} \ddot{\phi} &= -45.83\dot{\phi} - 414.53\phi - 277.32\phi_d \\ \ddot{\theta} &= -42.57\dot{\theta} - 252.15\theta - 183.62\theta_d \end{aligned} \tag{13}$$

Using this experimentally validated model and true system attitude dynamics, a sliding mode controller was designed.

III. Sliding Mode Controller

The goal of sliding mode control is to enforce the dynamics of the system to known desired trajectories. Theses trajectories are noted as x_d and $\dot{x}_d$ and the tracking errors are $\dot{\tilde{x}}$ = $\dot{x}$ - $\dot{x}_d$ and $\tilde{x}$ = x-x_d where x is a state. The sliding mode manifold is described by equation 14

$$\begin{aligned} s &= (\frac{d}{dt} + \lambda) \\ &= \dot{\tilde{x}} + \lambda\tilde{x} \end{aligned} \tag{14}$$

where $\lambda > 0$ is the slope of the sliding surface. A control law which causes s = 0 will asymptotically converge the tracking errors to zero.

Stability is demonstrated with a Lyapunov function V = $\frac{1}{2}s^2$ and $\dot{V}$ = $s\dot{s}$. The control input is designed to force $\dot{v} \leq$ -$\sigma|s|$ where $\sigma > 0$ and $\dot{s} = -\sigma$ sign(s). If an initial condition $x(t = 0)$ is not on the manifold, the system is forced to the manifold. Once on the manifold, the system dynamics are described by the equation for s. σ is chosen to be larger than the error between the estimated and actual system dynamics. When the system is operating near the manifold($s \approx 0$), noise in the observed quantities causes chatter to result from the function sign(s). The function is smoothed using the approximation sign(s) $\approx \frac{s}{|s|+\epsilon}$ where $\epsilon > 0$. When ϵ is made small (1e-3), control chatter is evident as sown in Figure 6.

For the simulation, the thrust input was calculated as show in equation 15 and the yaw input was calculated as in equation 16. Note that u_2 and u_3 were implemented using the characterization of the autopilot system with the inputs being the desired roll and pitch angles.

$$u_1 = \frac{m}{cos(\phi)cos(\theta)}(\ddot{z}_d - \lambda_z\dot{\tilde{z}} - \sigma_z\frac{s_z}{|s_z| + \epsilon} + g) \tag{15}$$

$$u_4 = I_z(\ddot{\psi}_d - \lambda_\psi\dot{\tilde{\psi}} - \sigma_\psi\frac{s_\psi}{|s_\psi| + \epsilon} + \frac{K_{faz}}{I_z}\dot{\psi}) \tag{16}$$

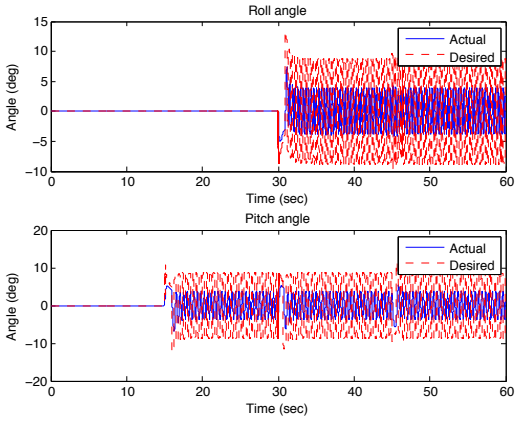

Fig. 6. Control Response with Chattering

When implementing the controller in hardware, yaw and thrust were controlled by a simple PID controller which has been experimentally verified to be accurate and reliable. This helped the tuning and debugging of the controller for x and y trajectory tracking.

Utilizing the onboard attitude controller, the sliding mode controller computes references angles to drive the translational errors to zero. The calculation of the desired pitch and roll angles are shown in equation 19 and 20.

$$u_x = \frac{m}{u_1}(\ddot{x}_d - \lambda_x \dot{\tilde{x}} - \sigma_x \frac{s_x}{|s_x| + \epsilon} + K_{ftx}\dot{x}) \quad (17)$$

$$u_y = \frac{m}{u_1}(\ddot{y}_d - \lambda_y \dot{\tilde{y}} - \sigma_y \frac{s_y}{|s_y| + \epsilon} + K_{fty}\dot{y}) \quad (18)$$

$$\phi_d = sin^{-1}(\frac{u_x sin(\psi) - u_y cos(\psi)}{cos(\psi) + sin(\psi)^2}) \quad (19)$$

$$\theta_d = sin^{-1}(\frac{u_y + sin(\phi_d)cos(\psi)}{cos(\phi_d)sin(\psi)}) \quad (20)$$

IV. Results

A. Simulated Results

Simulations were run using the values defined by Table I.

TABLE I

Simulation Parameter Values

Variable	Parameter	Value
m	Mass	0.5 kg
K_{fax} ,K_{fay},K_{faz}	Drag Coefficient (x,y,z)	$5 * 10^{-4}$ kg/s
K_{ftx} ,K_{fty},K_{ftz}	Drag Coefficient (x,y,z)	0.37 kg/s
d	Arm Length	0.25 m
I_x,I_y,I_z	Moment of Inertia (x,y,z)	0.005 $kg \cdot m^2$
g	Gravity	9.8 m/s^2
σ_x, σ_y, σ_ψ	Sliding Param.	1.5
σ_z	Sliding Param.	1
λ_x, λ_y, λ_ψ	Sliding Surface Slope	2
λ_z	Sliding Surface Slope	10
e	Smoothing Param.	.5

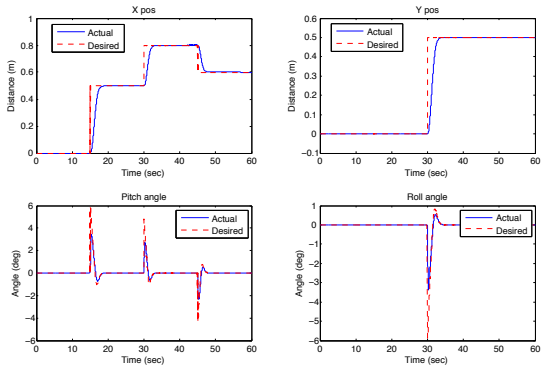

Fig. 7. Simulated Trajectory Following

The results of the control scheme are shown in Figure 7 where the red line is the desired position/attitude and the blue line is the simulated position/attitude. As mentioned previously, the autopilot response was used to model the plant in the simulations. A basic three-dimensional trajectory was given and the controller successfully maneuvered the quadrotor to follow the desired trajectory.

B. Experimental Results in Hardware

Once the controller was validated experimentally, it was implemented in hardware. The states were measured using the Vicon motion capture system running at 120 hz. The sliding mode controller computed reference roll and pitch angles while separate PID loops compute thrust and yaw commands. The controller proved effective at maintaining a stable hover at a desired position as shown in

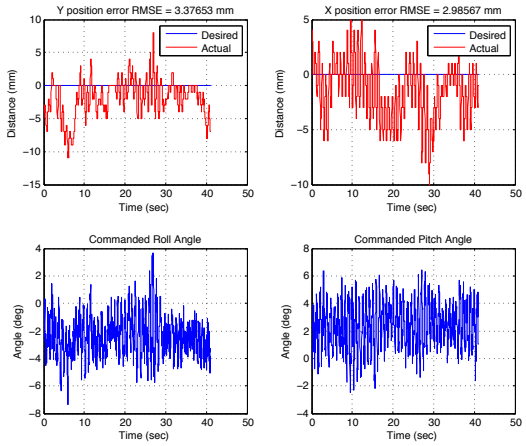

Fig. 8. Hovering Response

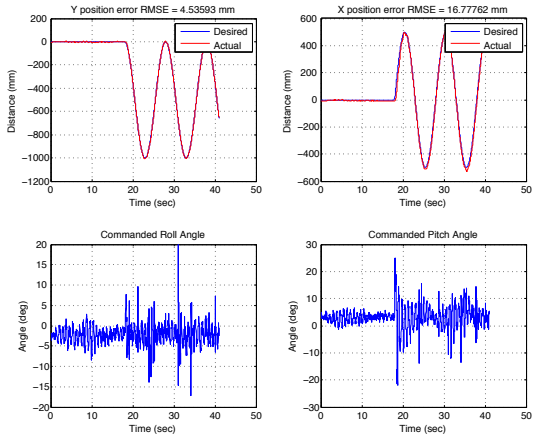

Fig. 9. Circle Trajectory Following Error

Figure 8 as well as following a circular trajectory as shown in Figure 10. When told to hover around a specific position, the root mean squared error (RMSE) was less than 3.5 mm in either axis as shown in Figure 8. When following the circular trajectory, the RMSE was 4.5 mm in the x axis and 16.8 mm in the y axis as shown in Figure 9. It is unclear why there is a disparity between the errors for each axis, but it is possible that further tuning could reduce the y axis error. The attached video shows both of these experiments.

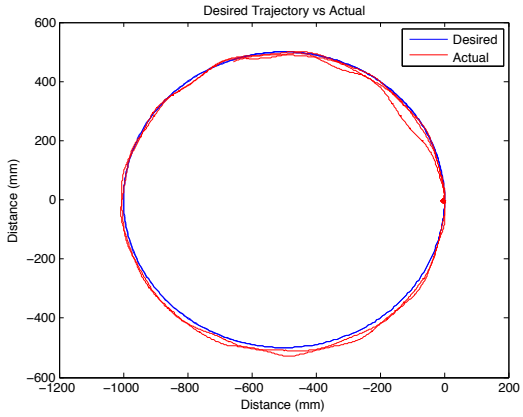

Fig. 10. Circle Trajectory Following

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

This study investigated a sliding mode controller to enable the quadrotor vehicle to follow a desired trajectory in the x and y axis. After deriving the basic model of the quadrotor, the model was validated experimentally and then adapted to represent the control scheme interface of the Ascending Technologies Hummingbird quadrotor. Step responses were perfomed to characterize the onboard autopilot system. Using this experimentally acquired data, the controller was designed to compute desired roll and pitch angles to position the quadrotor on the desired trajectory. Then controller was verified in simple simulations and then implemented in hardware to follow both static and dynamic desired trajectories. The controller proved accurate with small RMSE in each axis.

B. Future Works

In the future, the controller could be tuned and refined for improved performance. Since the reference angles were computed as function of the z axis error (u_1), there was some instability when this error became suddenly relatively large. It was unclear if this was a parameter tuning problem or if there is a better method of computing the desired roll and pitch values. The experiments relied on previously computed PID controllers for yaw

and altitude. These could be replaced so that the quadrotor is controlled entirely by sliding mode controllers.

REFERENCES

[1] J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice-Hall, 1991.

[2] H. Bouadi, M. Bouchoucha, and M. Tadjine, "Sliding mode control based on backstepping approach for an uav type-quadrotor," *International Journal of Applied Mathematics and Computer Sciences*, vol. 4, no. 1, pp. 12–17, 2008.

[3] L. Derafa, T. Madani, and A. Benallegue, "Dynamic modelling and experimental identification of four rotors helicopter parameters," in *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, dec. 2006, pp. 1834 –1839.

[4] B. C. Min, C. H. Cho, K. M. Choi, and D. H. Kim, "Development of a micro quad-rotor uav for monitoring an indoor environment," in *Proceedings of the FIRA RoboWorld Congress 2009 on Advances in Robotics*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 262–271.

[5] M. Efe, "Robust low altitude behavior control of a quadrotor rotorcraft through sliding modes," in *Control Automation, 2007. MED '07. Mediterranean Conference on*, june 2007, pp. 1 –6.

[6] B. Julian, "An embedded controller for quad-rotor flying robots running distributed algorithms," Master's thesis, Massachusetts Institute of Technology, United States of America, 2009.

[7] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, sept.-2 oct. 2004, pp. 2451 – 2456 vol.3.

[8] S. Bouabdallah, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *In Proceedings of IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 2247–2252.