

Fault Tolerant Super Twisting Sliding Mode Control of a Quadrotor UAV Using Control Allocation

Mehmet Karahan ^{a,1,*}, Mertcan Inal ^{a,2}, Cosku Kasnakoglu ^{a,3}

^a TOBB University of Economics and Technology, Electrical and Electronics Engineering Department, Sogutozu Street No: 43, Cankaya, Ankara 06510, Turkey

¹ m.karahan@etu.edu.tr; ² mertcan.inal@etu.edu.tr; ³ kasnakoglu@etu.edu.tr

* Corresponding Author

ARTICLE INFO

ABSTRACT

Article history

Received March 06, 2023

Revised April 19, 2023

Accepted April 21, 2023

Keywords

Actuator fault;

Altitude control;

Attitude control;

Fault-tolerant control;

Nonlinear control systems;

UAV Quadrotor;

Sliding mode control

In this study, a fault-tolerant super-twisting sliding mode controller with a control allocation system for a quadrotor aircraft is proposed. Super twisting sliding mode control is a robust control technique that handles a system with a relative degree equal to one. A super-twisting sliding mode controller is proposed because of its robustness to uncertainties and perturbations. It increases accuracy and reduces chattering. A control allocation algorithm is developed to cope with the actuator fault. Firstly, a nonlinear model of the quadrotor unmanned aerial vehicle (UAV) is presented. Then, the controller design and type of the actuator fault are explained. The control allocation algorithm is used to optimize the trajectory tracking performance of the quadrotor in the presence of an actuator fault. A control allocation algorithm is an effective approach to implementing fault-tolerant control. When actuator faults are identified, they can be modeled as changes in the B matrix of constraints. Various simulations have been made for situations with and without actuator failure. In normal conditions, the quadrotor can accurately track altitude, roll, pitch and yaw references. In faulty conditions, the quadrotor can follow the references with a small error. Simulations prove the effectiveness of the control allocation algorithm, which stabilizes the quadrotor in case of an actuator fault. Overall, this paper presents a novel fault-tolerant controller design for quadrotor aircraft that effectively addresses actuator faults using a super-twisting sliding mode controller and control allocation algorithm.

This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



1. Introduction

The first quadrotors began to be produced in the 1920s. Produced from the 1920s to the 1970s, quadrotors were large-scale aircraft controlled by a pilot. With the development of technology, quadrotor unmanned aerial vehicles have been produced since the late 1990s. While the first quadrotors used by the pilot were produced for the purpose of carrying people and cargo, quadrotor UAVs can be used for many different purposes [1-3]. In the last few years, the importance of quadrotor UAVs has increased due to their extensive range of applications, such as monitoring, firefighting, crop spraying, mapping, transporting goods, aerial photography and military operations. In general, quadrotor UAVs can be used in various civil operations and defense industry applications [4-7]. The wide range of use, low cost and simple mechanical structure of quadrotor unmanned aerial vehicles

have led many researchers to be interested in this issue [8]. Quadrotor unmanned aerial vehicles' vertical take-off and landing (VTOL) ability, hovering in the air, ability to rotate around their own axis, take-off and landing on uneven grounds make them more advantageous than winged unmanned aerial vehicles [9-12].

In practical applications, quadrotors are generally required to track a prescribed path accurately and rapidly. However, it is a challenging task because of parameter uncertainties, actuator faults and external disturbances [13]. Therefore, fault-tolerant controller design gains great importance for quadrotor unmanned aerial vehicles.

In the early studies, researchers mainly focused on PID controller design and linearized quadrotor dynamics, neglected some aerodynamic coefficients and ignored faults. Salih et al. designed a PID controller for a quadrotor UAV and neglected the inertia matrix of the aircraft and the gyroscopic effect caused by the propellers' rotation [14]. Ma et al. proposed a fuzzy PID controller for a quadrotor UAV, linearized the nonlinear system and neglected drag coefficients, aerodynamic, Coriolis and gyroscopic effects [15]. Shang et al. designed a fractional-order PID controller for a quadrotor UAV and ignored some small coefficients in the transfer function of the system [16]. Khatoon et al. compared PID and LQR controllers for a quadrotor UAV and linearized system about a certain operating point for using an LQR controller. It is concluded that the LQR controller is more robust and generates low steady-state error [17]. Huang et al. developed a backstepping controller, ignored some small coefficients related to inertia moments during the controller design process and compared results with a classical PID controller. It is observed that the backstepping controller has a lower settling time and faster time response than the PID controller [18]. Yu et al. developed an integral sliding mode controller for a quadrotor UAV and replaced the signum function with the saturation function to minimize the chattering effect in the controller's equations. Yu et al. also compared simulation results with the traditional PID controller and observed that the sliding mode controller converged rapidly to the attitude reference and showed better performance [19].

In the last few years, researchers have focused on fault-tolerant controller design for unmanned aerial vehicles. Fault-tolerant controllers maintain a stable performance regardless of disturbances [20], [21]. Ermeydan et al. designed a fault-tolerant PID controller for a quadrotor UAV. In the first and third rotors, a 20% loss of control effectiveness is simulated. It is observed that the PID controller can control the quadrotor stably up to 20% failure level in two rotors [22]. Sadeghzadeh et al. developed an active fault-tolerant PID controller for a quadrotor UAV, simulated a faulty scenario with up to 22% of power loss in power of all rotors and developed a gain scheduling strategy for faulty conditions [23]. Jun et al. proposed a fault-tolerant PID controller design for a quadrotor. It is observed that quadrotor flight is stable with a 2.2% height reduction when a 50% loss of effectiveness at one rotor [24]. Zhang et al. proposed linear quadratic regulator (LQR) and model predictive controllers (MPC) for a quadrotor UAV against rotor failures, considered 30% and 50% of loss effectiveness in one rotor, respectively and compared the results [25]. Nguyen et al. designed a fault-tolerant backstepping controller and performed simulations at 40% loss of control effectiveness at one rotor [26]. Some researchers have focused on developing a fault-tolerant sliding mode controller for the quadrotor UAV. Wang et al. developed an incremental sliding mode controller for a quadrotor UAV and simulated a 25% fault case for one rotor [27]. Yang et al. designed a sliding mode fault-tolerant controller for a quadrotor UAV with varying load and actuator faults. They proposed a method, a 40% power loss in one rotor at the 12th second [28]. Nguyen et al. developed a fault-tolerant sliding mode controller for a quadrotor UAV. They implemented a 40% loss of control effectiveness into the actuator [29]. In practical applications of sliding mode controllers, researchers may face the undesirable phenomenon of oscillations having finite frequency and amplitude, which is called chattering. Chattering leads to low control accuracy, high heat losses in power circuits and high wear of moving mechanical parts. It can arise from the fast dynamics which were neglected in the ideal model [30]. When a fault occurs in the system, the control allocation mechanism redistributes (re-allocates) the control signals produced by the controller among available actuators so that the

controller has the same effects on the system output or at least produces the same steady-state responses [31].

In this research, a fault-tolerant super-twisting sliding mode controller with control allocation is proposed for a quadrotor UAV. 60% loss of control effectiveness at one rotor is simulated. The control allocation algorithm is used to stabilize the controller in case of an actuator fault. MATLAB simulations are performed for the cases with and without rotor failure, and the difference is observed. The major contributions of this research are outlined as follows:

- 1) Fault-tolerant super-twisting sliding mode controller with a control allocation algorithm is developed to control the quadrotor in case of actuator fault. The sliding-mode controllers are more robust than the other controller methods described above, but they suffer from a chattering problem. Unlike classical sliding mode controllers, our proposed controller does not have a chattering problem.
- 2) Many researchers have studied actuator failure at rates ranging from 20% to 50%. Our study focused on an actuator failure rate of 60%, higher than many other studies. Thanks to the control allocation algorithm, the quadrotor is controlled in a stable manner despite the high rate of actuator failure.

The remainder of this paper is structured as follows. In [Section II](#), the dynamics of the quadrotor unmanned aerial vehicle are explained. In [Section III](#), the fault-tolerant super-twisting sliding mode control structure is explained. The MATLAB simulations of the proposed fault-tolerant sliding mode controller with control allocation are presented in [Section IV](#). Lastly, the conclusions of this research are stressed in [Section V](#).

2. Quadrotor Model

Quadrotor is a four-rotor unmanned aerial vehicle with vertical landing and take-off features. The power unit of the quadrotor consists of two opposing rotor pairs. The (1,3) rotor pair and (2,4) rotor pair rotate in opposite directions to each other. There are four different movements in the quadrotor UAV. Simultaneously increasing or reducing the speed of all four rotors of the quadrotor leads to vertical movement [32]. Changing the speed of the (1,3) rotor pair of the quadrotor creates roll motion. Raising or reducing the speed of the (2,4) rotor pair of the quadrotor creates the pitch motion. The difference in torque between the (1,3) and (2,4) rotor pairs produce the yaw motion. A schematic representation of the four-rotor unmanned aerial vehicle with 6 Degrees of Freedom (DOF) is given in [Fig. 1](#) [33]. The rotation direction of the rotors, the torques produced by the rotors, the Earth frame, the body frame, roll, pitch and yaw movements are represented in [Fig. 1](#).

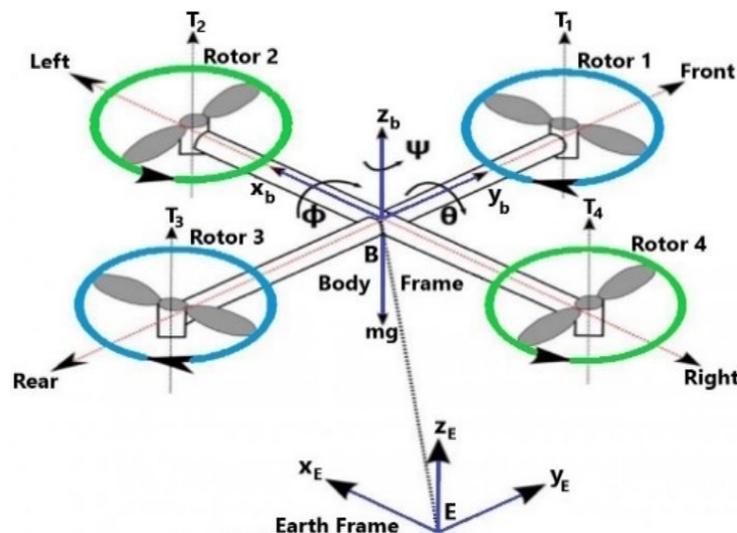


Fig. 1. Diagram of the quadrotor UAV

The quadrotor unmanned aerial vehicle with 6 DOF $[\varphi, \theta, \psi, x, y, z]$ consists of translational and rotational dynamics. Translational dynamics represent x, y, z axes, whereas rotational dynamics (φ, θ, ψ) present respectively roll, pitch and yaw. The Earth frame is described $E = [X_E, Y_E, Z_E]$ and the body frame is given as $B = [X_B, Y_B, Z_B]$ whose origin B defines the center of gravity in the quadrotor UAV [32]. The rotor is defined as a propulsion system, and it provides the thrust with a propeller [34]. In Fig. 2, the connection between the rotor and the propeller appears more clearly. Also, in Fig. 2, roll, pitch and yaw movements are shown on the axes [35].



Fig. 2. Rotor-propeller connection and rotational movements of the quadrotor UAV

Rotation matrix R is defined in (1), and it is used with the aim of transformation between Earth frame E and body frame B .

$$R = \begin{bmatrix} c\theta c\psi & s\varphi s\theta c\psi - c\varphi s\psi & c\varphi s\theta c\psi + s\varphi s\psi \\ c\theta s\psi & s\varphi s\theta s\psi + c\varphi c\psi & c\varphi s\theta s\psi - s\varphi c\psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \quad (1)$$

where

$$c = \cos, \quad s = \sin \quad (2)$$

I matrix represents the Inertia matrix and includes diagonal elements with their product equal to 0, having I_x, I_y, I_z moments of inertia. The matrix of Inertia is given in (3).

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (3)$$

The forces produced by rotors and moments have been explained by (4) and (5) as:

$$F_i = K_f \Omega_i^2 \quad (4)$$

$$M_i = K_m \Omega_i^2 \quad (5)$$

where K_f and K_m are constant coefficients, Ω_i^2 defines the i^{th} angular velocity. The actuator dynamics are represented in (6). U_1, U_2, U_3, U_4 are control inputs. F describes the total thrust produced by the propellers. $T_\varphi, T_\theta, T_\psi$, are torques along the respective axes. l presents the arm length of the quadrotor.

$$u = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} F \\ T_\varphi \\ T_\theta \\ T_\psi \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -lK_f & 0 & lK_f \\ lK_f & 0 & -lK_f & 0 \\ -K_m & K_m & -K_m & K_m \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (6)$$

The quadrotor's dynamical model is defined as

$$\ddot{\varphi} = \dot{\psi}\dot{\theta}\left(\frac{I_y - I_z}{I_x}\right) + \dot{\theta}\Omega_R \frac{J_R}{I_x} + \frac{l}{I_x} U_2 \quad (7)$$

$$\ddot{\theta} = \dot{\psi}\dot{\theta}\left(\frac{I_z - I_x}{I_y}\right) + \dot{\varphi}\Omega_R \frac{J_R}{I_x} + \frac{l}{I_y} U_3 \quad (8)$$

$$\ddot{\psi} = \dot{\varphi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{l}{I_z} U_4 \quad (9)$$

$$\ddot{X} = (\cos \varphi \sin \theta \cos \psi + \sin \psi \sin \varphi) \frac{U_1}{m} \quad (10)$$

$$\ddot{Y} = (\cos \varphi \sin \theta \sin \psi - \cos \psi \sin \varphi) \frac{U_1}{m} \quad (11)$$

$$\ddot{Z} = (\cos \theta \cos \varphi) \frac{U_1}{m} - g \quad (12)$$

where Ω_R and J_R are angular velocity and inertia moment of the propellers, m describes the quadrotor's mass, and g presents gravitational acceleration.

In this research, MATLAB simulations are performed using the physical properties of the Parrot AR Drone 2.0 quadrotor UAV. It has two interchangeable hulls. One is designed for indoors, and the other for outdoors. The hull designed for indoor use is heavier but more resistant to collisions. In this study, the indoor hull is used. Fig. 3 shows the Parrot AR Drone 2.0 with the indoor hull [36]. Table 1 represents the parameters of the Parrot AR Drone 2.0 with the indoor hull [37, 38].



Fig. 3. Parrot AR Drone 2.0 with indoor hull

Table 1. Quadrotor parameters

Symbol	Quantity	Value
m	Mass of the quadrotor	0.429 kg
l	Arm length of the quadrotor	0.1785 m
J_R	The inertia of each rotor	$2.03 \times 10^{-5} \text{ kg}\cdot\text{m}^2$
I_x	Inertia for x -axes	$2.24 \times 10^{-3} \text{ kg}\cdot\text{m}^2$
I_y	Inertia for y -axes	$2.98 \times 10^{-3} \text{ kg}\cdot\text{m}^2$
I_z	Inertia for z -axes	$4.80 \times 10^{-3} \text{ kg}\cdot\text{m}^2$
K_f	Thrust factor	$8.05 \times 10^{-6} \text{ N}/(\text{rad}/\text{s})^2$
K_m	Torque factor	$2.42 \times 10^{-7} \text{ N}\cdot\text{m}/(\text{rad}/\text{s})^2$
Ω_{max}	Maximum speed	1047.2 rad/s

3. Controller Design

The fault-tolerant super twisting sliding mode control is proposed to track the given altitude and attitude references in case of 60% actuator failure at one rotor. The control allocation algorithm is activated when one of the four rotors fails and enables the quadrotor to follow the references stably.

3.1. Super Twisting Sliding Mode Controller

The sliding mode controller design is more robust than other controller methods in the literature. Two stages are very important in the sliding mode controller approach.

- 1) The selection of the sliding surface
- 2) State variables must remain on the sliding surface

The sliding phase and the reaching phase of the sliding mode controller are represented in Fig. 4. and the sliding surface is defined in (13) [39].

$$s = \dot{e} + \lambda e \quad (13)$$

where e is the error and λ represents the tuning parameter.

The sliding variables used in the sliding mode controller are given in (14).

$$s^T = [\varphi \ \theta \ \psi \ z] \quad (14)$$

The sliding variables must reach the sliding surface in finite time and slide on the surface to reach 0 asymptotically with the aim of controlling the quadrotor to the reference trajectory [40].

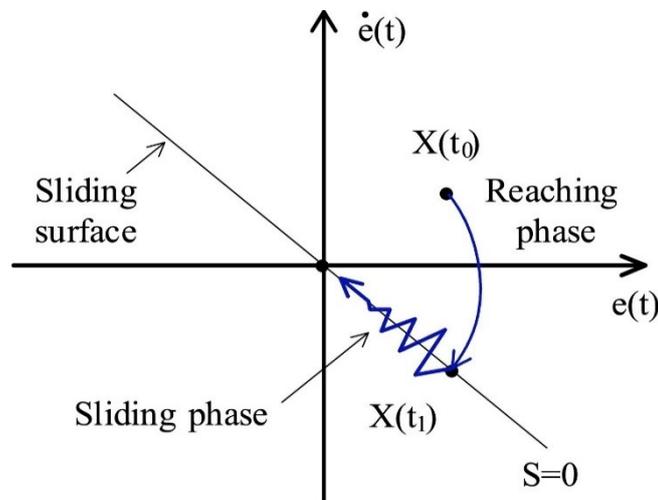


Fig. 4. Sliding phase and reaching phase

The equivalent control inputs for the sliding mode controller can be summarized as follows.

$$U_{1eq} = \frac{m}{\cos \varphi \cos \theta} (g + \ddot{z}_d - \lambda(\dot{z}_d - \dot{z})) \quad (15)$$

$$U_{2eq} = \frac{I_x}{l} (-\dot{\theta} \dot{\psi} \left(\frac{I_y - I_z}{I_x} \right) + \ddot{\varphi} - \lambda(\dot{\varphi}_d - \dot{\varphi})) \quad (16)$$

$$U_{3eq} = \frac{I_y}{l} (-\dot{\varphi} \dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) + \ddot{\theta} - \lambda(\dot{\theta}_d - \dot{\theta})) \quad (17)$$

$$U_{4eq} = \frac{I_z}{l} (-\dot{\varphi} \dot{\theta} \left(\frac{I_x - I_y}{I_z} \right) + \ddot{\psi}_d - \lambda(\dot{\psi}_d - \dot{\psi})) \quad (18)$$

A discontinuous term is added to the equivalent control for the purpose of satisfying the sliding reachability condition by substituting

$$\dot{s} = -\text{sign}(s) \quad (19)$$

The equivalent control equation can be written as follows.

$$U_1 = U_{1eq} + \text{sign}(s_z) \quad (20)$$

$$U_2 = U_{2eq} + \text{sign}(s_\varphi) \quad (21)$$

$$U_3 = U_{3eq} + \text{sign}(s_\theta) \quad (22)$$

$$U_4 = U_{4eq} + \text{sign}(s_\psi) \quad (23)$$

The sign function being discontinuous will cause a high-frequency oscillation called the chattering effect [41]. In order to cope with the chattering effect sign function is replaced with an alternative continuous function (24).

$$\text{fun}(s) = [(1 + s)^n - (1 - s)^n] / [(1 + s)^n + (1 - s)^n] \quad (24)$$

where n is an even integer greater than 0, signum function and alternative continuous function graphs are shown in Fig. 5 through Fig. 6. Fig. 5 represents the alternative continuous function when $n = 2$.

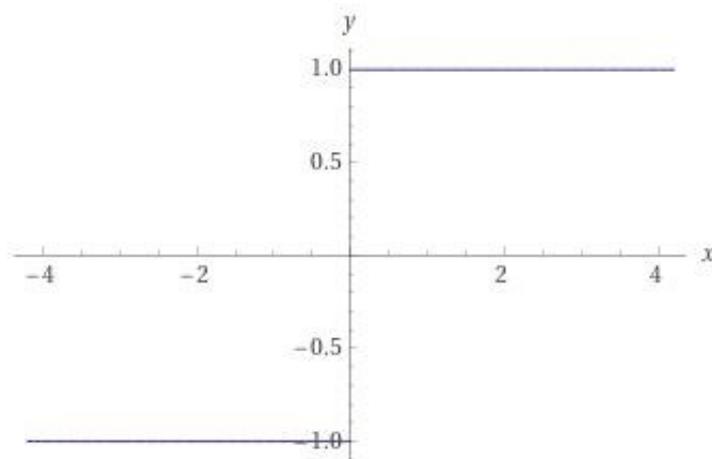


Fig. 5. The signum function

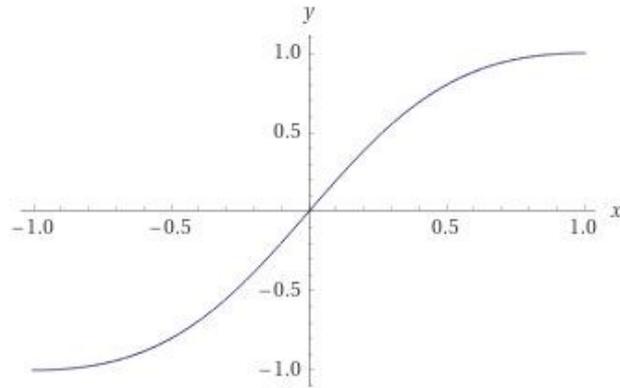


Fig. 6. A continuous function with $n = 2$

The use of the continuous function reduces chattering; however, it can lead to decreasing the robustness of the controller. A higher-order linear switching function can be replaced instead of a nonlinear switching function with the purpose of reducing the chattering effect. Super-twisting sliding mode controller presents such a switching function that can balance with the robustness of the controller and high-frequency chattering.

In order for the sliding variables to reach the sliding surface in a finite time and without chattering, the following higher-order switching function is used.

$$\dot{S} = 1.5k^{1/2}s^{1/2} \text{sign}(s) + \int 1.1k \text{sign}(s) \quad (25)$$

The control inputs for the super-twisting sliding mode controller are calculated in Equations (26)-(29) where k_1, \dots, k_4 are positive constants.

$$U_1 = U_{1eq} - 1.5k_1^{1/2} s_z^{1/2} \text{sign}(s_z) - \int 1.1k_1 \text{sign}(s_z) \quad (26)$$

$$U_2 = U_{2eq} - 1.5k_2^{1/2} s_\varphi^{1/2} \text{sign}(s_\varphi) - \int 1.1k_2 \text{sign}(s_\varphi) \quad (27)$$

$$U_3 = U_{3eq} - 1.5k_3^{1/2} s_\theta^{1/2} \text{sign}(s_\theta) - \int 1.1k_3 \text{sign}(s_\theta) \quad (28)$$

$$U_4 = U_{4eq} - 1.5k_4^{1/2} s_\psi^{1/2} \text{sign}(s_\psi) - \int 1.1k_4 \text{sign}(s_\psi) \quad (29)$$

3.1. Control Allocation Algorithm

When one of the rotors loses its effectiveness by 60%, the controller should counter the loss of effectiveness and continue the reference track safely. The loss of actuator effectiveness is calculated by comparing the desired angular velocity of the rotor (w_d) with the actual angular velocity of the rotor (w). The loss of effectiveness (LE) is calculated as in (30).

$$LE = (w_d - w)/w_d \quad (30)$$

The LE value is always between 0 and 1. When the LE value is 0, there is no power loss in the rotor, and when it is 1, there is a 100% power loss in the rotor. When the LE value rises above 0, the control allocation algorithm is activated, and the power loss in the rotor is compensated. When one of the propellers of the quadrotor fails, the quadrotor loses its balance. When the faults are determined,

the angular velocity of the rotor opposite to the faulty rotor is calculated such that the torque balance is maintained. The control effectiveness of the quadrotor UAV is defined in (31).

$$k = 1 - LE \quad (31)$$

The control effectiveness vector is described in (32).

$$K = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix} \quad (32)$$

where $k_i \in [0, 1]$ and $k_i = 1$ means that the actuator is completely functional and $k_i = 0$ describes the case where the actuator does not work at all. After the fault is detected, the control output is represented as

$$\begin{bmatrix} F \\ T_\phi \\ T_\theta \\ T_\psi \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -lK_f & 0 & lK_f \\ lK_f & 0 & -lK_f & 0 \\ -K_m & K_m & -K_m & K_m \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (33)$$

The fault-tolerant super twisting sliding mode controller diagram is represented in Fig. 7. Sliding mode control is a variable structure control method. The state feedback control law is not a continuous function of time. It can switch from one continuous structure to another continuous structure based on the current position in the state space [42], [43]. The sliding variable must be used to define the sliding surface. The reference trajectory is the trajectory given for the quadrotor to follow. The quadrotor tries to follow this trajectory using the sliding mode controller. When an actuator error occurs, and power loss occurs in one of the rotors, the control allocation mechanism is activated. It readjusts the power distribution between the rotors and keeps the quadrotor flying stably.

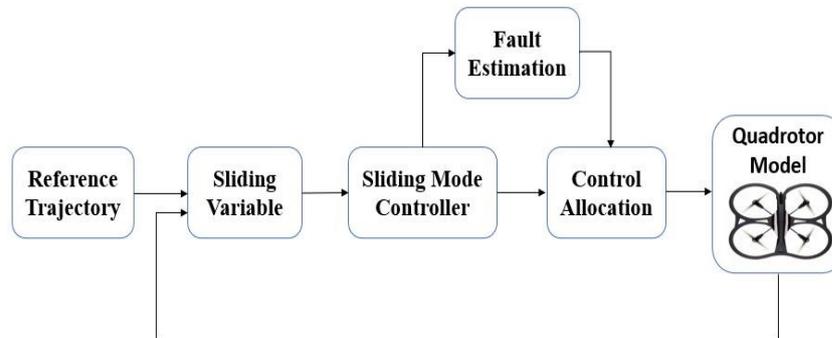


Fig. 7. Block scheme of the proposed controller

4. Result and Discussions

In this part, the efficiency of the proposed fault-tolerant super-twisting sliding mode controller is verified by MATLAB simulations. The physical parameters of the quadrotor defined in the second section are used in the MATLAB simulations. An actuator fault due to 60% power loss in the fourth rotor is introduced at the beginning of the simulation. Simulations were made separately for cases with and without actuator failure. In this way, it has been observed more clearly how tolerant the proposed controller is against actuator errors. Fig. 8-11 shows the roll, pitch, yaw and altitude tracking responses of the controller without actuator failure.

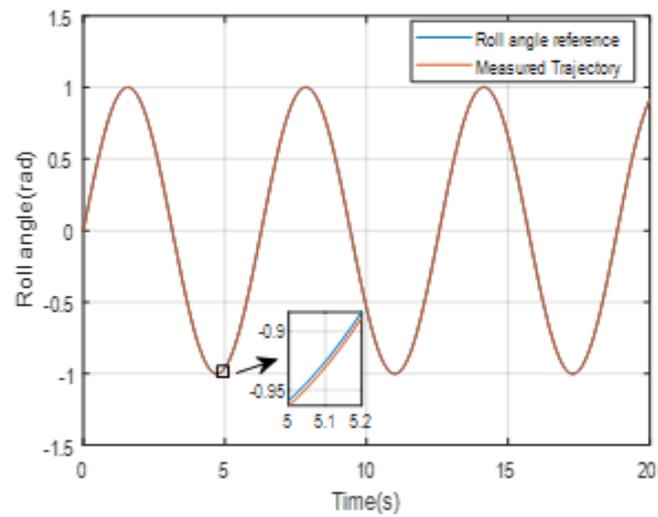


Fig. 8. Roll angle reference tracking simulation of the controller without rotor failure

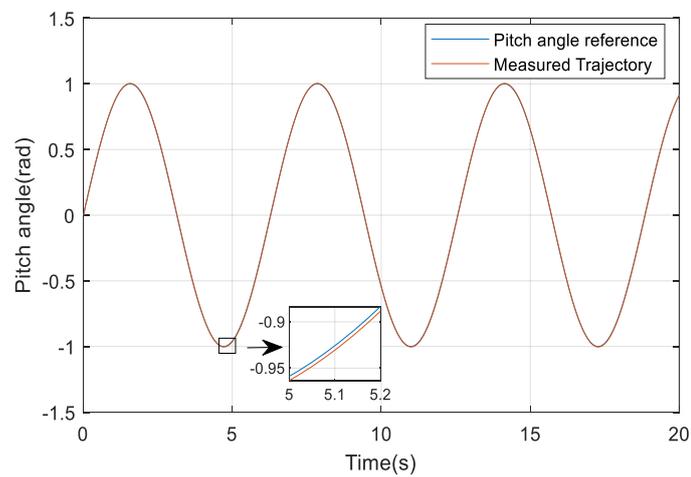


Fig. 9. Pitch angle reference tracking simulation of the controller without rotor failure

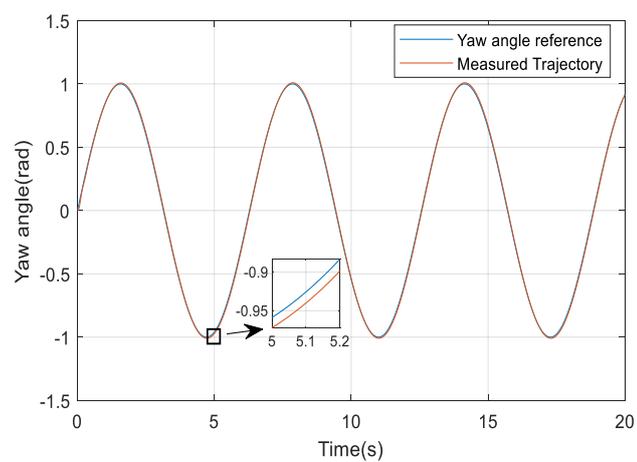


Fig. 10. Yaw angle reference tracking simulation of the controller without rotor failure

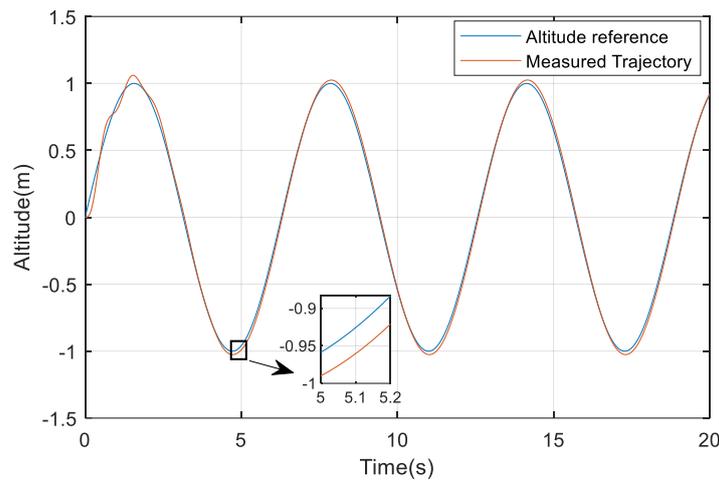


Fig. 11. Altitude reference tracking simulation of the controller without rotor failure

It is clear from the simulations in Fig. 8-11 that the quadrotor can follow the given sinusoidal references with almost 0 error. The references for roll, pitch, and yaw angles are in radians, and the reference for altitude is in meters. The red line in the graphs represents the given sinusoidal reference, while the blue line shows the trajectory followed by the quadrotor. The part between 5 and 5.2 seconds in the graphs has been enlarged to provide a clearer observation of the reference (red line) and the trajectory followed by the quadrotor (blue line).

Table 2 gives the time response (rise time, overshoot and settling time) values of the controller without rotor failure. Rise time is defined as the time required for a signal to change from 10% to 90% of a specified value. Overshoot is the occurrence of a signal exceeding its target. Settling time is the time required by the response to reach and steady within a specified range of 2 % of its final value. When the data in Table 2 is carefully examined, it is seen that the references of roll, pitch and yaw angles are followed with a slight overshoot of around 0.5%. The altitude reference is followed by an overshoot of 2.2%. The controller followed the roll, pitch and yaw references very closely from the start of the simulation and never left the $\pm 2\%$ range. Therefore, the settling time is 0 seconds for these references. The altitude reference shows a little more overshoot and settles in the $\pm 2\%$ range in 0.25 seconds. In the absence of rotor error, the controller can follow the given references almost without error.

Table 2. Time response of the controller without rotor failure

Reference	Rise time (s)	Overshoot (%)	Settling time (s)
Roll angle	1.828	0.503	0
Pitch angle	1.828	0.503	0
Yaw angle	1.829	0.505	0
Altitude	1.818	2.2	0.25

Fig. 12-15 show simulations indicating the reference tracking results when the fourth rotor of the quadrotor experiences 60% power loss. In case of a rotor failure in the quadrotor, the control allocation algorithm activates, ensuring that the quadrotor stably follows the given references. The control allocation algorithm can compensate for the failure of the fourth rotor with the help of the other 3 rotors.

In case of 60% rotor failure, the quadrotor can still follow the given references. However, a small overshoot is observed at the peaks of the sinusoidal references. This is due to power loss in one rotor of the quadrotor. As explained in Section II, the torque difference between the rotor pairs (1,3) and (2,4) produces the yaw motion. Due to the power loss in the fourth rotor, the torque difference between the rotor pairs increases, and a higher overshoot occurs during the yaw angle reference tracking. When

evaluated in general, it is clear that the control allocation algorithm works successfully and achieves stable reference tracking by compensating for rotor failure.

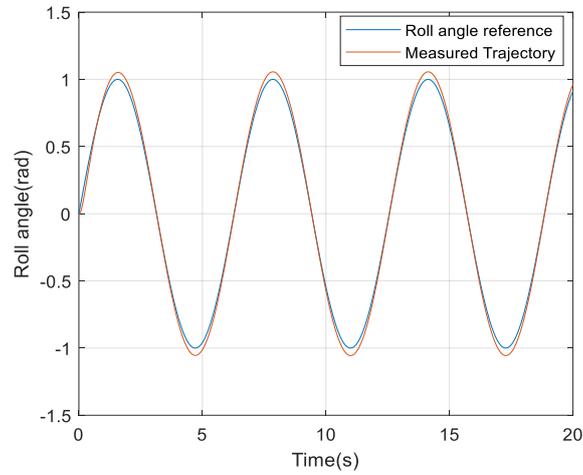


Fig. 12. Roll angle reference tracking simulation of the controller with rotor failure

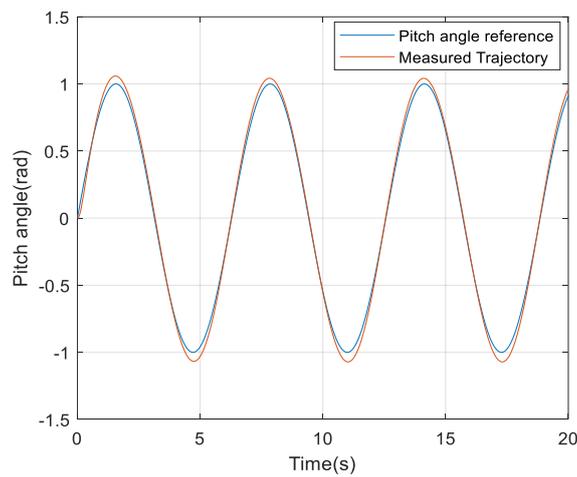


Fig. 13. Pitch angle reference tracking simulation of the controller with rotor failure

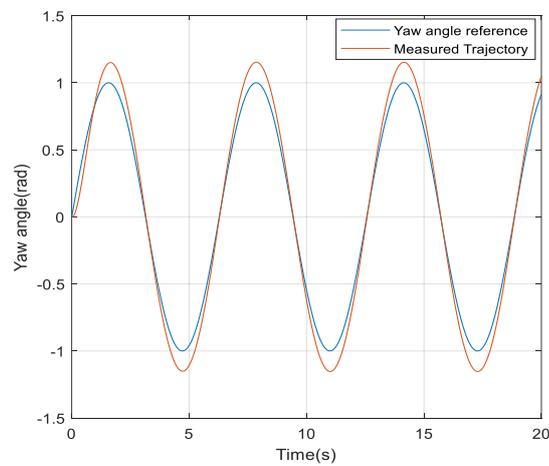


Fig. 14. Yaw angle reference tracking simulation of the controller with rotor failure

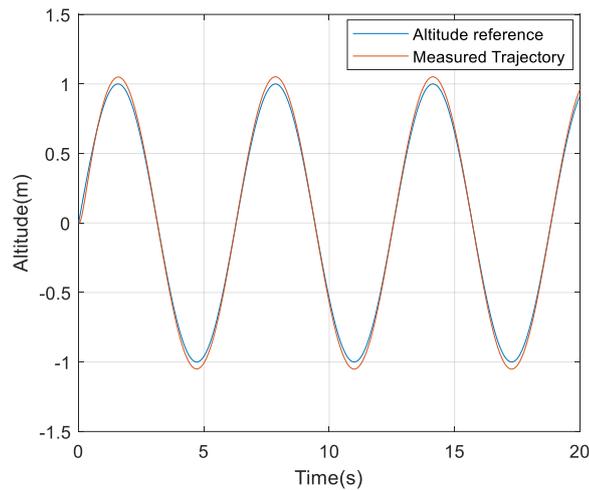


Fig. 15. Altitude reference tracking simulation of the controller with rotor failure.

Table 3 gives the time response (rise time, overshoot and settling time) values of the controller with rotor failure. When the time response data in Table 3 is examined, it is seen that the rotor error increases the overshoot. However, it is not high enough to prevent the quadrotor from flying stably. The settling time is slightly longer than when there is no rotor error. However, the reference value is captured at around 0.6 seconds for roll, pitch and altitude references. The yaw reference can be captured in about 1 second. The system has a short settling time despite rotor failure.

Table 3. Time response of the controller with rotor failure

Reference	Rise time (s)	Overshoot (%)	Settling time (s)
Roll angle	1.829	5.70	0.65
Pitch angle	1.807	5.72	0.60
Yaw angle	1.830	15.2	1.08
Altitude	1.829	4.80	0.64

5. Conclusion

This research proposes a fault-tolerant super-twisting sliding mode controller for a quadrotor. Firstly, the development of the quadrotor's dynamic model is explained. Then, the working principle of the proposed controller is explained. To eliminate the chattering effect in the sliding mode controller, the signum function has been replaced with a continuous function. Then, it is explained how the fault-tolerant controller is implemented by using the control allocation algorithm. Thanks to the control allocation algorithm, 60% power loss in a single rotor can be tolerated. In this way, the quadrotor can follow the given references in a fault-tolerant manner. In the simulations section, different simulations have been made for cases with and without rotor failure. In this way, the efficiency of the fault-tolerant sliding mode controller is better observed. Simulations were made for reference tracking, and the results were compared. It has been observed that in the absence of rotor failure, the quadrotor follows the sinusoidal references almost flawlessly. In the case of 60% rotor failure in a single rotor, it has been observed that the quadrotor follows the references stably thanks to the control allocation algorithm. In addition, rise time, overshoot and settling time data were obtained for cases with or without rotor failure. When the data obtained are examined, it is seen that the quadrotor successfully follows all the references, almost without overshoot, in the absence of rotor failure. When power loss occurs due to rotor error, the quadrotor can follow the references by showing a small overshoot. As a result, it has been observed that the fault-tolerant super-twisting sliding mode controller with control allocation works successfully. In future studies, a control allocation algorithm will be developed that will enable the quadrotor aircraft to fly stably in case of a failure of the two rotors.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

Acknowledgment: We would like to express our sincere gratitude to our supervisor, Professor Cosku Kasnakoglu, for his valuable guidance and support throughout the research process. His expertise and insights were invaluable in shaping our research and helping us to overcome challenges.

Conflicts of Interest: The authors declare no conflict of interest.

References

- [1] S. N. Ghazbi, Y. Aghli, M. Alimohammadi, and A. A. Akbari, "Quadrotors unmanned aerial vehicles: A review," *Int. J. Smart Sens. Intell. Syst.*, vol. 9, no. 1, pp. 309-333, 2016, <https://doi.org/10.21307/ijssis-2017-872>.
- [2] J. Kim, S. A. Gadsden, and S. A. Wilkerson, "A Comprehensive Survey of Control Strategies for Autonomous Quadrotors," *Canadian Journal of Electrical and Computer Engineering*, vol. 43, no. 1, pp. 3-16, 2020, <https://doi.org/10.1109/CJECE.2019.2920938>.
- [3] T. Oktay and Y. Eraslan, "Numerical Investigation of Effects of Airspeed and Rotational Speed on Quadrotor UAV Propeller Thrust Coefficient," *J. Aviat.*, 2021, <https://doi.org/10.30518/jav.872627>.
- [4] A. Saggai, S. Zeghlache, L. Benyettou, and A. Djerioui, "Fault Tolerant Control against Actuator Faults Based on Interval Type-2 Fuzzy Sliding Mode Controller for a Quadrotor Aircraft," *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, 2021, pp. 181-186, <https://doi.org/10.1109/IHSH51661.2021.9378754>.
- [5] R. Akbar and N. Uchiyama, "Adaptive modified super-twisting control for a quadrotor helicopter with a nonlinear sliding surface," *2017 SICE International Symposium on Control Systems (SICE ISCS)*, 2017, pp. 1-6, <https://ieeexplore.ieee.org/document/7889630>.
- [6] J. Zheng and P. Li, "Fault Tolerant Control of Actuator Additive Fault for Quadrotor Based on Sliding Mode Observer," *2019 Chinese Control and Decision Conference (CCDC)*, 2019, pp. 5908-5913, <https://doi.org/10.1109/CCDC.2019.8833258>.
- [7] M. Karahan and C. Kasnakoglu, "Path Planning and Collision Avoidance with Artificial Intelligence for a Quadrotor UAV," *2021 International Conference Automatics and Informatics (ICAI)*, 2021, pp. 163-166, <https://doi.org/10.1109/ICAI52893.2021.9639807>.
- [8] X. -L. Ren, "Observer Design for Actuator Failure of a Quadrotor," *IEEE Access*, vol. 8, pp. 152742-152750, 2020, <https://doi.org/10.1109/ACCESS.2020.3017522>.
- [9] M. N. Shauqee, P. Rajendran, and N. M. Suhadis, "Quadrotor Controller Design Techniques and Applications Review," *INCAS Bulletin*, vol. 13, no. 3, pp. 179-194, 2021, <https://doi.org/10.13111/2066-8201.2021.13.3.15>.
- [10] V. S. Chipade, Abhishek, M. Kothari, and R. R. Chaudhari, "Systematic design methodology for development and flight testing of a variable pitch quadrotor biplane VTOL UAV for payload delivery," *Mechatronics*, vol. 55, pp. 94-114, 2018, <https://doi.org/10.1016/j.mechatronics.2018.08.008>.
- [11] K. K. Deveerasetty, Y. Zhou, and B. Han, "Trajectory tracking of a quadrotor using TID controller," *2019 Australian & New Zealand Control Conference (ANZCC)*, 2019, pp. 139-144, <https://doi.org/10.1109/ANZCC47194.2019.8945770>.
- [12] P. Patel and J. Dave, "Design and Dynamic Modelling of Quadrotor VTOL aircraft," *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2020, pp. 105-112, <https://doi.org/10.1109/ICECA49313.2020.9297379>.
- [13] N. P. Nguyen, N. X. Mung, H. L. N. N. Thanh, T. T. Huynh, N. T. Lam and S. K. Hong, "Adaptive Sliding Mode Control for Attitude and Altitude System of a Quadcopter UAV via Neural Network," *IEEE Access*, vol. 9, pp. 40076-40085, 2021, <https://doi.org/10.1109/ACCESS.2021.3064883>.
- [14] L. Salih, M. Moghavvemi, H. A. F. Mohamed, and K. S. Gaeid, "Modelling and PID controller design for a quadrotor unmanned air vehicle," *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 2010, pp. 1-5, <https://doi.org/10.1109/AQTR.2010.5520914>.
- [15] J. Ma and R. Ji, "Fuzzy PID for Quadrotor Space Fixed-Point Position Control," *2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, 2016, pp. 721-726, <https://doi.org/10.1109/IMCCC.2016.131>.

- [16] B. Shang, Y. Zhang, C. Wu, and Y. Chen, "Fractional Order Flight Control of Quadrotor UAS: an OS4 Benchmark Environment and a Case Study," *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 377-382, <https://doi.org/10.1109/ICARCV.2018.8581196>.
- [17] S. Khatoun, D. Gupta, and L. K. Das, "PID & LQR control for a quadrotor: Modeling and simulation," *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2014, pp. 796-802, <https://doi.org/10.1109/ICACCI.2014.6968232>.
- [18] T. Huang and D. Huang, "Backstepping control for a quadrotor unmanned aerial vehicle," *2020 Chinese Automation Congress (CAC)*, 2020, pp. 2475-2480, <https://doi.org/10.1109/CAC51589.2020.9326765>.
- [19] H. Yu, S. Wu, Q. Lv, Y. Zhou, and S. Liu, "Robust integral sliding mode controller for quadrotor flight," *2017 Chinese Automation Congress (CAC)*, 2017, pp. 7352-7656, <https://doi.org/10.1109/CAC.2017.8244106>.
- [20] C. Zhang and I. M. Jaimoukha, "Fault-tolerant controller design with a tolerance measure for systems with actuator and sensor faults," *2017 American Control Conference (ACC)*, 2017, pp. 4129-4134, <https://doi.org/10.23919/ACC.2017.7963589>.
- [21] M. Özkan and Ç. Kasnakoglu, "Active Fault Detection in Linear Controller Hardware with Sine Signal," *2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)*, 2020, pp. 90-94, <https://doi.org/10.1109/ICEEE49618.2020.9102484>.
- [22] A. Ermeydan and E. Kiyak, "Fault tolerant control against actuator faults based on enhanced PID controller for a quadrotor," *Aircraft Engineering and Aerospace Technology*, vol. 89, no. 3, pp. 468-476, 2017, <https://doi.org/10.1108/AEAT-04-2015-0096>.
- [23] I. Sadeghzadeh, A. Mehta, A. Chamseddine, and Y. Zhang, "Active Fault Tolerant Control of a quadrotor UAV based on gainscheduled PID control," *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2012, pp. 1-4, <https://doi.org/10.1109/CCECE.2012.6335037>.
- [24] W. Jun, Y. Xiong-Dong, and T. Yu-Yang, "Fault-Tolerant Control Design of Quadrotor UAV Based on CPSO," *2018 IEEE 4th International Conference on Control Science and Systems Engineering (ICCSSE)*, 2018, pp. 279-283, <https://doi.org/10.1109/CCSSE.2018.8724803>.
- [25] B. Yu, Y. Zhang, I. Minchala, and Y. Qu, "Fault-tolerant control with linear quadratic and model predictive control techniques against actuator faults in a quadrotor UAV," *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, 2013, pp. 661-666, <https://doi.org/10.1109/SysTol.2013.6693925>.
- [26] N. P. Nguyen and S. K. Hong, "Active Fault-Tolerant Control of a Quadcopter against Time-Varying Actuator Faults and Saturations Using Sliding Mode Backstepping Approach," *Applied Sciences-Basel*, vol. 9, no. 19, pp. 3390-4010, Sept. 2019. <https://doi.org/10.3390/app9194010>.
- [27] X. Wang, S. Sun, E.-J. van Kampen, and Q. Chu, "Quadrotor fault tolerant incremental sliding mode control driven by sliding mode disturbance observers," *Aerospace Science and Technology*, vol. 87, pp. 417-430, 2019, <https://doi.org/10.1016/j.ast.2019.03.001>.
- [28] P. Yang P, Z. Wang, Z. Zhang, and X. Hu, "Sliding Mode Fault Tolerant Control for a Quadrotor with Varying Load and Actuator Fault," *Actuators*, vol. 10, no. 12, pp. 323-340, 2021, <https://doi.org/10.3390/act10120323>.
- [29] N. P. Nguyen and S. K. Hong, "Active fault-tolerant control of a quadcopter against time-varying actuator faults and saturations using sliding mode backstepping approach," *Applied Sciences*, vol. 9, no. 19, pp. 4010, 2019, <https://doi.org/10.3390/app9194010>.
- [30] V. Utkin and Hoon Lee, "Chattering Problem in Sliding Mode Control Systems," *International Workshop on Variable Structure Systems*, 2006, pp. 346-350, <https://doi.org/10.1109/VSS.2006.1644542>.
- [31] A. Argha, S. W. Su, Y. Liu, and B. G. Celler, "Control Allocation Based Sliding Mode Fault Tolerant Control," *2019 American Control Conference (ACC)*, 2019, pp. 3752-3757, <https://doi.org/10.23919/ACC.2019.8814769>.
- [32] Z. Liu, X. Liu, J. Chen, and C. Fang, "Altitude Control for Variable Load Quadrotor via Learning Rate Based Robust Sliding Mode Controller," *IEEE Access*, vol. 7, pp. 9736-9744, 2019, <https://doi.org/10.1109/ACCESS.2018.2890450>.
- [33] M. Karahan and C. Kasnakoglu, "Nonlinear Modeling and Lyapunov Stability Based Controller Design of a Quadrotor UAV," *2022 3rd International Informatics and Software Engineering Conference (IISEC)*, 2022, pp. 1-6, <https://doi.org/10.1109/IISEC56263.2022.9998100>.
- [34] İ. Gov, "Rotor spacing and blade number effect on the thrust, torque and power of a coaxial rotor," *El-Cezeri*, vol. 7, no. 2, pp. 487-502, 2020, <https://doi.org/10.31202/ecjse.660136>.
- [35] L. O. Rojas-Perez and J. Martinez-Carranza, "DeepPilot: A CNN for autonomous drone racing," *Sensors*, vol. 20, no. 16, pp. 4524-4544, 2020, <https://doi.org/10.3390/s20164524>.

- [36] K. S. Lee, M. Ovinis, T. Nagarajan, R. Sculin, and O. Morel, "Autonomous patrol and surveillance system using unmanned aerial vehicles," *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, 2015, pp. 1291-1297, <https://doi.org/10.1109/EEEIC.2015.7165356>.
- [37] I. Viana, L. Santana, D.A. dos Santos, and L. Goes, "Experimental Validation of a Trajectory Tracking Control using the AR.Drone Quadrotor," *Congresso Brasileiro de Engenharia Mecânica (CONEM)*, 2016, pp. 1444-1453, <https://www.researchgate.net/publication/307471911>.
- [38] B. E. Demir, R. Bayir, and F. Duran, "Real-time trajectory tracking of an unmanned aerial vehicle using a self-tuning fuzzy proportional integral derivative controller," *Int. J. Micro Air Vehicles*, vol. 8, no. 4, pp. 252-268, 2016, <https://doi.org/10.1177/1756829316675882>.
- [39] W. Alqaisi, J. Ghommam, A. Alazzam, M. Saad, and V. Ner-guizian, "Three-loop uncertainties compensator and sliding mode quadrotor control," *Computers & Electrical Engineering*, vol. 81, pp. 106507, 2020, <https://doi.org/10.1016/j.compeleceng.2019.106507>.
- [40] L. Besnard, Y. B. Shtessel, and B. Landrum, "Control of a Quadrotor Vehicle Using Sliding Mode Disturbance Observer," *2007 American Control Conference*, 2007, pp. 5230-5235, <https://doi.org/10.1109/ACC.2007.4282421>.
- [41] X. Lin and L. Wang, "Heading control of Air Cushion Vehicle with disturbance observer based on terminal sliding mode," *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2017, pp. 858-862, <https://doi.org/10.1109/ICMA.2017.8015928>.
- [42] V. Utkin, A. Poznyak, Y. Orlov, and A. Polyakov, "Conventional and high order sliding mode control," *J. Frank. Inst.*, vol. 357, no. 15, pp. 10244-10261, 2020, <https://doi.org/10.1016/j.jfranklin.2020.06.018>.
- [43] V. Utkin, A. Poznyak, Y. V. Orlov, and A. Polyakov, *Road Map for Sliding Mode Control Design*, Springer Cham, 2020, <https://doi.org/10.1007/978-3-030-41709-3>.