drones

MDPI

Article

Finite-Time Neuro-Sliding-Mode Controller Design for Quadrotor UAVs Carrying Suspended Payload

Özhan Bingöl [1,*] and Hacı Mehmet Güzey [2,†]

[1] Department of Electrical and Electronics Engineering, Gumushane University, Baglarbası Str., 29100 Gümüşhane, Turkey
[2] Department of Electrical and Electronics Engineering, Sivas University of Science and Technology, Kardesler Str. No: 7/1, 58100 Sivas, Turkey
* Correspondence: ozhan.bingol@gumushane.edu.tr
† These authors contributed equally to this work.

Abstract: Due to the quadrotor's underactuated nature, suspended payload dynamics, parametric uncertainties, and external disturbances, designing a controller for tracking the desired trajectories for a quadrotor that carries a suspended payload is a challenging task. Furthermore, one of the most significant disadvantages of designing a controller for nonlinear systems is the infinite-time convergence to the desired trajectory. In this paper, a finite-time neuro-sliding mode controller (FTNSMC) for a quadrotor with a suspended payload that is subject to parametric uncertainties and external disturbances is designed. By constructing a finite-time sliding mode controller, the quadrotor can follow the reference trajectories in finite time. Furthermore, despite time-varying nonlinear dynamics, parametric uncertainties, and external disturbances, a neural network structure is added to the controller to effectively reduce chattering phenomena caused by high switching gains, and significantly reduce the size of the control signals. Following the completion of the controller design, the system's stability is demonstrated using the Lyapunov stability criterion. Extensive numerical simulations with various scenarios are run to demonstrate the effectiveness of the proposed controller.

Keywords: finite-time stability; neural network; quadrotor UAV; sliding mode control

Citation: Bingöl, Ö.; Güzey, H.M. Finite-Time Neuro-Sliding-Mode Controller Design for Quadrotor UAVs Carrying Suspended Payload. *Drones* **2022**, *6*, 311. https://doi.org/10.3390/drones6100311

Academic Editors: Samir Khan, Vaios Lappas, Nadjim Horri and Andrey V. Savkin

Received: 3 October 2022
Accepted: 19 October 2022
Published: 21 October 2022

1. Introduction

Unmanned system research is expanding quickly as a result of recent technology developments that have led to lower costs and smaller equipment [1]. The most popular of these technologies is unmanned aerial vehicles (UAVs), which have a larger range of civil and military applications [2,3]. The quadrotor is a type of UAV that shows great potential and is recognized as an ideal UAV by most research studies because of features such as simple construction, excellent maneuverability, and low cost [4]. Studies on UAVs and load-carrying applications are quickly expanding as a result of their widespread deployment. Due to its great maneuverability, ability to take off and land vertically, and capacity to carry payloads almost as heavy as its own body weight, the quadrotor is an excellent choice for autonomous transportation [5]. There are numerous ways to attach loads to quadrotor UAVs, including using a robotic arm to pick up the weight [6], mounting it to the quadrotor [7], and attaching it to the quadrotor using a cable [8,9]. The final technique, a quadrotor with a payload, offers a number of benefits over the first two: the construction is simpler, there are fewer restrictions on the size and form of the cargo, and it does not require landing for loading or unloading, which can save time and energy throughout the transport process. Moreover, using a robotic arm or attaching the payload directly to the quadrotor can severely limit the quadrotor's ability to land vertically.

Due to the aforementioned benefits, the relevance of carrying a suspended payload is increased by its usage in civil applications, including the delivery of first-aid supplies to disaster zones, cargo delivery, and agricultural spraying [5].

Despite its advantages, the suspended load-carrying quadrotor system is nonlinear, strongly coupled, and underactuated [10]. Additionally, unknown external disturbances and parametric uncertainties degrade the system's control performance. Considering these difficulties, developing a powerful controller for the the suspended load-carrying quadrotor system is difficult.

1.1. Related Work

Different control strategies were proposed in the literature to control a quadrotor payload system. In [11], a PID controller was proposed, feedback linearization-based controller was established in [9], and a backstepping-based nonlinear controller was presented in [12]. To actively regulate the position of the load, [8] suggested a nonlinear geometric control technique. In [13], the construction of a PID-based geometric controller was described that allowed for the payload to asymptotically follow a predetermined trajectory for both the payload position and attitude. In order to resolve the challenges brought on by the intense interaction of the system states and underactuated features of the dynamic system, in [14], the controller design was split into two parts, the UAV's attitude control design, and the UAV's position control and payload's swing motion control design. The second part's control rule, which takes into account the suppression of the payload's swing motion, was developed using the partial feedback linearization process. With an underactuated quadrotor payload system, wind disturbances were also considered in [15]. A path-following controller based on an uncertainty and disturbance estimator was suggested. In addition to wind disturbances, load variations were also taken into account in [16]. To deal with transient disturbance and estimate the system characteristics, an adaptive robust controller for dynamic subsystems was designed. For the kinematic subsystem, a global sliding mode controller is used to produce the necessary attitude angles for following the intended 3D trajectory.

The above control strategies are founded on theories of exponential or asymptotic stability. In reality, obtaining rapid tracking may not be possible with such approaches. To overcome this, a finite-time control strategy was devised for several systems in the literature [17]. For a class of nonstrict feedback nonlinear systems that contain input saturation, unidentified smooth functions, and error limitations, the challenge of finite-time control using neural networks is studied in [18]. The stochastically finite-time control problem for uncertain multiple-input, multiple-output (MIMO) stochastic nonlinear systems is addressed in [19] using a nontriangular solution. In [20], a finite-time control system using the nonsingular fast terminal sliding mode (NFTSM) and finite-time disturbance observer (FDO) approaches is developed to tackle the precise trajectory tracking issue of a surface vehicle affected by complicated marine environments. The adaptive finite-time decentralized control issue for nonlinear large-scale systems with time-varying output constraints and input saturation is discussed in [21]. According to the aforementioned research, finite-time control is a practical and efficient approach that, in theory, ensures the controlled system's rapid transient response and resilience and exactly satisfies the demand for the transient and robust performance of nonlinear systems.

1.2. Contributions of This Work

The sliding mode controller (SMC) is a robust control method that is regularly applied to nonlinear systems due to its benefits, such as its insensitivity to parameter variations, the avoidance of external disturbances, and quick dynamic responses [22]. Despite these advantages, it is extremely difficult to design an efficient controller due to drawbacks such as the chattering effect in the control signal, the requirement to accept uncertainties within a certain range, and the controller's adaptability to significant parameter changes and external disturbances [23]. Many different methods were developed in the literature to successfully eliminate this phenomenon [24–26]. With the recent developments in the area of artificial intelligence, artificial-intelligence-based controllers are being developed for different nonlinear systems in order to eliminate these disadvantages of SMC [27–31]. Any

function may be approximated using neural networks [32,33]. It can both estimate the nonlinearity of the systems and lower the amplitude of the SMC's chattering.

In a previous work [34], we designed a novel neural-network-based SMC to control the quadrotor payload system and eliminate the disadvantages of SMC. In order to improve our previous work by using the above-mentioned advantages regarding finite-time stability, a finite-time neuro-sliding mode controller structure for quadrotor UAV carrying a suspended payload is proposed. The main contributions of this study are listed as follows:

1. Compared to many studies in the literature [10,35,36], a comprehensive nonlinear mathematical model of the quadrotor payload system was established by taking into account external disturbances and parameter uncertainties.
2. A novel finite-time neuro-sliding mode controller design is proposed for a quadrotor transporting a suspended payload. The proposed controller also takes into account suspended payload dynamics, unlike the system discussed in [37]. It also has a neural-network component when compared to the study in [38]. Thus, a more comprehensive control structure was obtained compared to systems using only SMC as in [39]. While the proposed controller uses the robust structure of the SMC, it also successfully learns the unknown dynamics with the help of the neural-network component, overcomes the disadvantages of the SMC, unlike in studies such as [40], and ensures that the system states reach the desired trajectory, or their errors reach zero or a close neighbourhood of zero in finite time. Without finite-time analysis, stability is shown mathematically when time goes to infinity; we show that neuro-sliding mode control converges in finite time to be able to use it in practice.
3. Comprehensive stability analysis with Lyapunov stability theory for the proposed controller is derived to demonstrate the payload-carrying quadrotor's finite-time stability.

The paper is structured as follows. Section 2 provides a basic overview of quadrotor dynamics, payload dynamics, neural networks, and finite-time stability. In Section 3, the suggested neural-network-based finite-time sliding-mode controller for quadrotor UAV carrying suspended payload is established, and the efficacy of the technique is demonstrated using numerical simulations in Section 4. Section 5 concludes with some remarks.

Next, the background and preliminaries are given.

2. Theoretical Background and Preliminaries

This section provides a theoretical overview of quadrotor dynamics, payload dynamics, finite-time stability, and neural networks. Moreover, some useful definitions and lemmas are given to construct the proposed controller properly.

2.1. Preliminaries

In this section, the following definitions and lemmas are provided to properly model the proposed control structure in a later section.

Definition 1 ([41]). *Nonlinear system $\dot{x} = f(x,u)$ is semiglobal practical finite-time stable (SGPFS) in equilibrium point $x = 0$ if there is a constant $\mu > 0$ with $x(t_0) = x_0$ and settling time $T(\mu, x_0) < \infty$ ensuring $\|x(t)\| < \mu$, for all $t > t_0 + T$.*

Lemma 1 ([41]). *Nonlinear system $\dot{x} = f(x,u)$ is SGPFS if there is a positive definitive function $V(x)$ and constants $c > 0$, $0 < \lambda < 1$, and $\delta > 0$ that ensure:*

$$\dot{V}(x) \leq -cV^{\lambda}(x) + \delta, t \geq 0 \tag{1}$$

Reaching time is given for $\forall 0 < \gamma \leq 1$ as follows:

$$T_r = \frac{1}{(1-\lambda)\gamma c}\left(V^{1-\lambda}(x(0)) - \left(\frac{\delta}{(1-\gamma)c}\right)^{\frac{1-\lambda}{\lambda}}\right) \tag{2}$$

Lemma 2 ([42]). *For $\zeta_i \in \Re$, $i = 1, ..., n$, $0 < p < 1$, the following inequality holds:*

$$\left(\sum_{i=1}^{n} |\zeta_i|\right)^p \leq \sum_{i=1}^{n} |\zeta_i|^p \leq n^{1-p}\left(\sum_{i=1}^{n} |\zeta_i|\right)^p \tag{3}$$

Lemma 3 ([42]). *The following Young's inequality is true for $x, y \geq 0$ and $p, q > 1$ such that $\frac{1}{p} + \frac{1}{q} = 1$:*

$$xy \leq \frac{x^p}{p} + \frac{y^q}{q} \tag{4}$$

Lemma 4 ([42]). *The following inequality holds where τ, v are real variables and $a, b, c > 0$:*

$$|\tau|^a |v|^b \leq \frac{a}{a+b} c|\tau|^{a+b} + \frac{b}{a+b} c^{\frac{-a}{b}} |v|^{a+b} \tag{5}$$

2.2. Quadrotor Dynamics

The dynamic equations of a quadrotor are established in this section with the following assumptions [43]:

1. The quadrotor's construction is rigid and symmetrical.
2. The center of gravity of the quadrotor corresponds with the origin of the body axis.
3. The propellers are rigid.
4. Thrust and drag forces are related to the squares of the propeller's speed.
5. The UAV's roll and pitch angles are presumed to be operated within a range of $(-\pi/2, \pi/2)$, and yaw angle within $(-\pi, \pi)$.

A rotor on each arm of the quadrotor controls four fundamental movements that allow for it to achieve the required position and attitude. Four input signals are specified for the quadrotor's four motions. The initial input is u_1, which changes the speed of all propellers in the same proportion. u_1 controls the vertical movement of the quadrotor in the z axis. The second input signal, u_2, causes a torque along the body's x axis by changing the speed of the left propeller while also changing the speed of the right propeller by an equal amount. As a result, roll movement along the x axis obtained. The third input, u_3, which uses the same concept as u_2, adjusts the speed of the propellers on the front and rear rotors at the same time, leading in pitching motion along the y axis. The fourth input, u_4, simultaneously decreases or raises the speed of the left and right propellers while raising or reducing the speed of the rear and front propellers. As a result, yaw movement is achieved, which is the quadrotor turning around its own axis.

Because the rotors were symmetrically positioned, gyroscopic effects and aerodynamic torques balance each other out in flight, as seen in Figure 1. Two frames of reference are specified to express the quadrotor's position and attitude. The first frame of reference is the Earth frame, which is indicated by $E\{x, y, z\}$ and represents the position of the vehicle's center of gravity. The other is the fixed body frame, which is represented by $B\{x, y, z\}$.

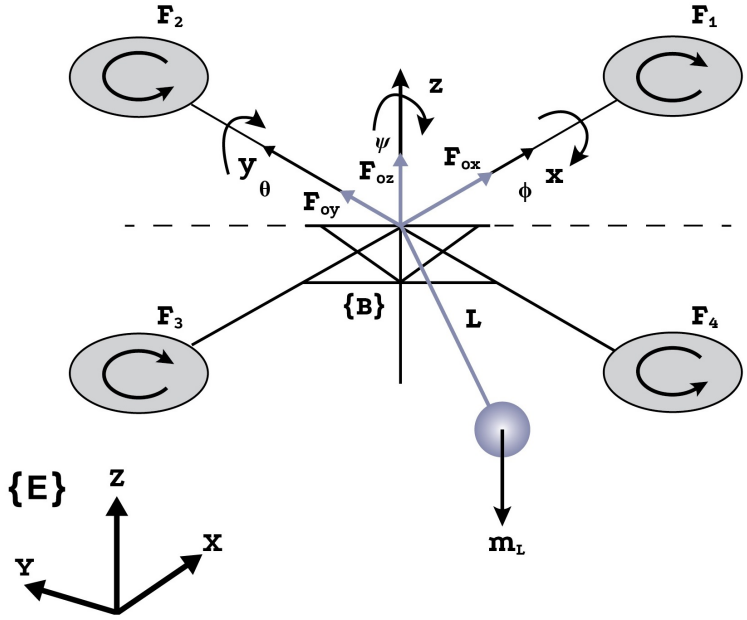

Figure 1. Quadrotor model with payload.

The dynamic quadrotor UAV model may be formed with parametric uncertainties and unknown external disturbances, and the model is as follows [44,45]:

$$\begin{aligned}
\ddot{x} &= \frac{1}{m}(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)u_1 - \frac{K_1\dot{x}}{m+\Delta m} + D_x \\
\ddot{y} &= \frac{1}{m}(\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi)u_1 - \frac{K_2\dot{y}}{m+\Delta m} + D_y \\
\ddot{z} &= \frac{1}{m}(\cos\phi\cos\theta)u_1 - g - \frac{K_3\dot{z}}{m+\Delta m} + D_z \\
\ddot{\phi} &= \dot{\theta}\dot{\psi}\frac{I_y+\Delta I_y - I_z - \Delta I_z}{I_x - \Delta I_x} + \frac{J_r}{I_x-\Delta I_x}\Omega_r\dot{\theta} + \frac{l}{I_x-\Delta I_x}u_2 - \frac{K_4 l}{I_x-\Delta I_x}\dot{\phi} + D_\phi \\
\ddot{\theta} &= \dot{\psi}\dot{\phi}\frac{I_z+\Delta I_z - I_x - \Delta I_x}{I_y - \Delta I_y} + \frac{J_r}{I_y-\Delta I_y}\Omega_r\dot{\phi} + \frac{l}{I_y-\Delta I_y}u_3 - \frac{K_5 l}{I_y-\Delta I_y}\dot{\theta} + D_\theta \\
\ddot{\psi} &= \dot{\phi}\dot{\theta}\frac{I_x+\Delta I_x - I_y - \Delta I_y}{I_z - \Delta I_z} + \frac{1}{I_z-\Delta I_z}u_4 - \frac{K_6}{I_z-\Delta I_z}\dot{\psi} + D_\psi
\end{aligned} \tag{6}$$

where (ϕ, θ, ψ) are Euler angles indicating roll, pitch and yaw angles; (x, y, z) are positions of the UAV in the space; m is the UAV's total mass; g is gravity acceleration; l is the space from the center of the UAV to the propellers; (I_x, I_y, I_z) are inertias with respect to axes; Jr is the inertia of the propeller; $K_i(i = 1, 2, \ldots, 6)$ are the drag coefficients; $\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$, Ω_i are the angular velocities of the propellers, $u_i(i = 1, 2, 3, 4)$ are the virtual inputs given in (7); $D_i(i = x, y, z, \phi, \theta, \psi)$ are external disturbances; Δ denotes parametric uncertainties.

$$\begin{aligned}
u_1 &= (F_1 + F_2 + F_3 + F_4) \\
u_2 &= (-F_2 + F_4) \\
u_3 &= (-F_1 + F_3) \\
u_4 &= \frac{d(-F_1 + F_2 - F_3 + F_4)}{b}
\end{aligned} \tag{7}$$

where $F_i(i = 1, 2, 3, 4)$ are rotor thrusts, b is the lift coefficient, d is the force-to-moment scaling vector.

2.3. Payload Dynamics

In this study, the effects of the load on the quadrotor in three dimensions are dynamically modeled and added to the quadrotor's own dynamics, and the total dynamic equations of the system are obtained. The payload model was developed on the following assumptions:

1. The payload may not rotate along the cable axis; therefore, payloads have only two degrees of freedom in tilting directions.
2. The load is suspended from a weightless, nonelastic cable, which is rigid.
3. The cable junction was placed at the quadrotor's centre of mass, and the tensile force of the cable does not directly impact the quadrotor's rotating motion.

As seen in Figure 2, where the payload was modeled as a three-dimensional point pendulum mass, the equations defining the dynamics of the load were derived by taking into account the α_x longitudinal suspension angles in the $x-z$ plane and the α_y longitudinal angles and forces in the $y-z$ plane. The following are the forces generated by the load on the respective axes [46]:

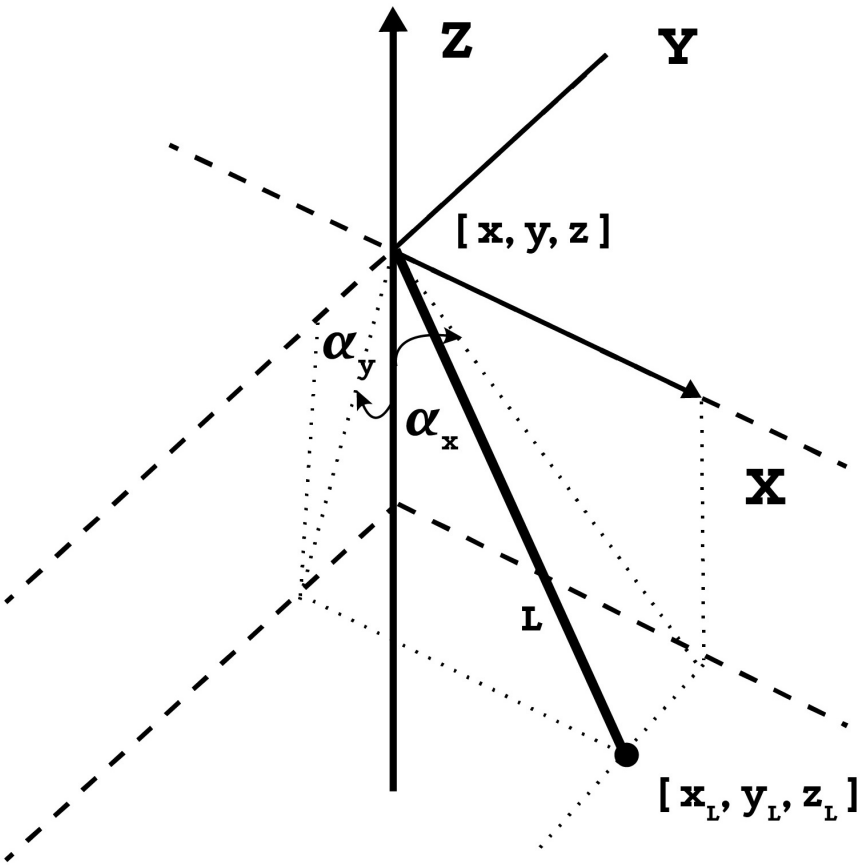

Figure 2. Point mass payload model.

$$\begin{aligned} F_{ox} &= m_L L\left(\ddot{\alpha}_x \cos\alpha_y \cos\alpha_x - \dot{\alpha}_x^2 \cos\alpha_y \sin\alpha_x\right) \\ F_{oy} &= m_L L\left(\ddot{\alpha}_y \cos\alpha_x \cos\alpha_y - \dot{\alpha}_y^2 \cos\alpha_x \sin\alpha_y\right) \\ F_{oz} &= m_L L\ddot{\alpha}_x \cos\alpha_y \sin\alpha_x + m_L L\dot{\alpha}_x^2 \cos\alpha_y \cos\alpha_x \\ &\quad + m_L L\ddot{\alpha}_y \cos\alpha_x \sin\alpha_y + m_L L\dot{\alpha}_y^2 \cos\alpha_x \cos\alpha_y - m_L g \end{aligned} \tag{8}$$

where m_L is the payload's mass; F_{ox}, F_{oy}, F_{oz} are the payload's forces effect on quadrotor in three-dimensional space shown in Figure 2.

Hence, the entire system dynamics are as follows:

$$\begin{aligned}
\ddot{x} &= \frac{1}{(m+m_L)}(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi)u_1-\frac{K_1\dot{x}}{(m+m_L+\Delta m)}-\frac{F_{ox}}{(m+m_L)}+D_x\\
\ddot{y} &= \frac{1}{(m+m_L)}(\cos\phi\sin\theta\sin\psi+\sin\phi\cos\psi)u_1-\frac{K_2\dot{y}}{(m+m_L+\Delta m)}-\frac{F_{oy}}{(m+m_L)}+D_y\\
\ddot{z} &= \frac{1}{(m+m_L)}(\cos\phi\cos\theta)u_1-\frac{mg}{(m+m_L)}-\frac{K_3\dot{z}}{(m+m_L+\Delta m)}+\frac{F_{oz}}{(m+m_L)}+D_z\\
\ddot{\phi} &= \dot{\theta}\dot{\psi}\frac{I_y+\Delta I_y-I_z-\Delta I_z}{I_x-\Delta I_x}+\frac{J_r}{I_x-\Delta I_x}\Omega_r\dot{\theta}+\frac{l}{I_x-\Delta I_x}u_2-\frac{K_4 l}{I_x-\Delta I_x}\dot{\phi}+D_\phi\\
\ddot{\theta} &= \dot{\psi}\dot{\phi}\frac{I_z+\Delta I_z-I_x-\Delta I_x}{I_y-\Delta I_y}+\frac{J_r}{I_y-\Delta I_y}\Omega_r\dot{\phi}+\frac{l}{I_y-\Delta I_y}u_3-\frac{K_5 l}{I_y-\Delta I_y}\dot{\theta}+D_\theta\\
\ddot{\psi} &= \dot{\phi}\dot{\theta}\frac{I_x+\Delta I_x-I_y-\Delta I_y}{I_z-\Delta I_z}+\frac{1}{I_z-\Delta I_z}u_4-\frac{K_6}{I_z-\Delta I_z}\dot{\psi}+D_\psi
\end{aligned} \tag{9}$$

2.4. Neural Network

In this work, a two-layer neural network (NN) structure was used to estimate uncertain dynamics as shown in Figure 3.

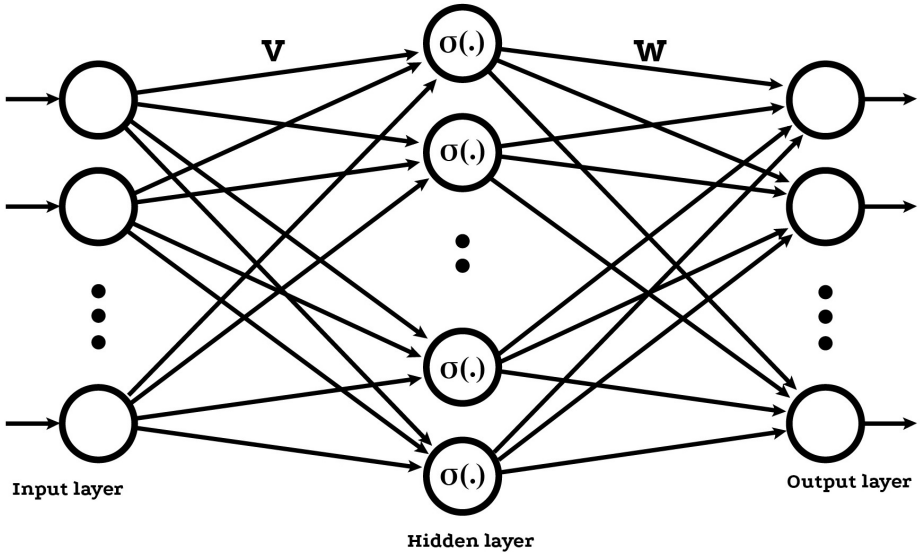

Figure 3. Neural-network structure.

The first layer is hidden and contains adjustable hidden weights, $W \in \mathbb{R}^{(Vh\times ko)}$, and the second layer consists of randomly determined constants, $\hat{V} \in \mathbb{R}^{(ki\times Vh)}$ where ki is the input count and ko is the output count. Vh represents the number of neurons in the hidden layer. Estimation function $f(x)$ can be given as $f(x) = W^T\sigma(\hat{V}^T\bar{x}) + \varepsilon$ where ε is the bounded NN estimation error that satisfies $\|\varepsilon\| < \varepsilon_M$, and $\sigma(\bullet) : \mathbb{R}^a \to \mathbb{R}^L$ is the activation function for the hidden layer. Because input-layer weights $\hat{V} \in \mathbb{R}^{ki\times Vh}$ are randomly chosen, for any input x, the estimation is viable; hence, activation function, $\sigma(x) = \sigma(\hat{V}^T\bar{x})$, establishes a stochastic base in compact set S [47]. $\bar{x}^T = [b_i \;\; x^T]$, where $b_i > 0$ is a bias term of input that lets threshold value to be in weight matrix $\hat{V}$. As an activation function, a tangent hyperbolic function in the form of $\sigma(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$, was chosen in this study for all neurons. It was also assumed that target weights were limited to a known positive value W_M satisfying $\|W\|_F < W_M$ on any compact subset of $\mathbb{R}^n$ [48]. Moreover, $\|\bullet\|$ and $\|\bullet\|_F$ are considered as the vector and Frobenius norm, respectively [47].

Next, the controller design is given.

3. Controller Design

The suggested controller is outlined here. The proposed controller takes into consideration quadrotor dynamics, and the unpredictable dynamics caused by external disturbances and parametric uncertainties. To reduce the chattering influence of the SMC, the unknown dynamics are learned using two-layer NN. Because of the neural-network component, the controller can successfully govern the system without understanding the quadrotor dynamics. Furthermore, utilizing the updated learning structure, system dynamics may be appropriately updated even when there are uncertainties that change over time. With the assistance of a finite-time control system, the trajectory errors converge to zero in a finite time.

To enable effective controller design, the dynamic equations obtained in (9) are subjected to the following transformation [45]:

$$\begin{cases} \ddot{z} = f_1 + g_1 u_1 + \Delta f_1 \\ \begin{cases} \ddot{x} = f_2 + g_2 u_2 + \Delta f_2 \\ \ddot{\phi} = f_3 + g_3 u_2 + \Delta f_3 \end{cases} \\ \begin{cases} \ddot{y} = f_4 + g_4 u_3 + \Delta f_4 \\ \ddot{\theta} = f_5 + g_5 u_3 + \Delta f_5 \end{cases} \\ \ddot{\psi} = f_6 + g_6 u_4 + \Delta f_6 \end{cases} \tag{10}$$

where

$$\begin{aligned} f_1 &= -\frac{mg}{(m+m_L)} - \frac{K_3 \dot{z}}{(m+m_L)} + \frac{F_{oz}}{(m+m_L)} + D_z \\ \Delta f_1 &= \frac{\Delta m K_3 \dot{z}}{(m+m_L)(m+\Delta m+m_L)}, g_1 = \frac{\cos\phi\cos\theta}{(m+m_L)} \\ f_2 &= \frac{(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi)u_1}{(m+m_L)} - \frac{K_1\dot{x}}{(m+m_L)} - \frac{F_{ox}}{(m+m_L)} + D_x \\ \Delta f_2 &= \frac{\Delta m K_1 \dot{x}}{(m+m_L)(m+\Delta m+m_L)}, g_2 = 0 \\ f_3 &= \frac{\left(\dot{\theta}\dot{\psi}(I_y-I_z) + J_r\dot{\theta}\Omega_r - K_4 l\dot{\phi}\right)}{I_x} + D_\phi, g_3 = \frac{l}{I_x} \\ \Delta f_3 &= \frac{-\left(\dot{\theta}\dot{\psi}(I_y-I_z) + J_r\dot{\theta}\Omega_r - K_4 l\dot{\phi}\right)\Delta I_x}{I_x(\Delta I_x+I_x)} + \dot{\theta}\dot{\psi}\left(\frac{\Delta I_y - \Delta I_z}{\Delta I_x + I_x}\right) \\ f_4 &= \frac{(\cos\phi\sin\theta\sin\psi+\sin\phi\cos\psi)u_1}{(m+m_L)} - \frac{K_2\dot{y}}{m+m_L} - \frac{F_{oy}}{m+m_L} + D_y \\ \Delta f_4 &= \frac{\Delta m K_2 \dot{y}}{(m+m_L)(m+\Delta m+m_L)}, g_4 = 0 \\ f_5 &= \frac{\left(\dot{\psi}\dot{\phi}(I_z-I_x) + J_r\dot{\phi}\Omega_r - K_5 l\dot{\theta}\right)}{I_y} + D_\theta, g_5 = \frac{l}{I_y} \\ \Delta f_5 &= \frac{-\left(\dot{\psi}\dot{\phi}(I_z-I_x) + J_r\dot{\phi}\Omega_r - K_5 l\dot{\theta}\right)\Delta I_y}{I_y(\Delta I_y+I_y)} + \dot{\psi}\dot{\phi}\left(\frac{\Delta I_z - \Delta I_x}{\Delta I_y + I_y}\right) \\ f_6 &= \frac{\left(\dot{\phi}\dot{\theta}(I_x-I_y) - K_6\dot{\psi}\right)}{I_z} + D_\psi, g_6 = \frac{1}{I_z} \\ \Delta f_6 &= \frac{-\left(\dot{\phi}\dot{\theta}(I_x-I_y) - K_6\dot{\psi}\right)\Delta I_z}{I_z(\Delta I_z+I_z)} + \dot{\psi}\dot{\phi}\left(\frac{\Delta I_x - \Delta I_y}{\Delta I_z + I_z}\right) \end{aligned} \tag{11}$$

Moreover, sliding surface functions are selected as follows [49,50]:

$$
\begin{aligned}
s_1 &= a_{11}(\dot{z}_d - \dot{z}) + \int_0^t (a_{12} sign^{\alpha_1}(z_d - z) + a_{11} sign^{\alpha_2}(\dot{z}_d - \dot{z}))dt \\
s_2 &= a_{21}(\dot{x}_d - \dot{x}) + a_{23}(\dot{\phi}_d - \dot{\phi}) + \int_0^t (a_{22} sign^{\alpha_1}(x_d - x) + a_{21} sign^{\alpha_2}(\dot{x}_d - \dot{x}))dt \\
&+ \int_0^t (a_{24} sign^{\alpha_1}(\phi_d - \phi) + a_{23} sign^{\alpha_2}(\dot{\phi}_d - \dot{\phi}))dt \\
s_3 &= a_{31}(\dot{y}_d - \dot{y}) + a_{33}\left(\dot{\theta}_d - \dot{\theta}\right) + \int_0^t (a_{32} sign^{\alpha_1}(y_d - y) + a_{31} sign^{\alpha_2}(\dot{y}_d - \dot{y}))dt \\
&+ \int_0^t \left(a_{34} sign^{\alpha_1}(\theta_d - \theta) + a_{33} sign^{\alpha_2}\left(\dot{\theta}_d - \dot{\theta}\right)\right)dt \\
s_4 &= a_{41}(\dot{\psi}_d - \dot{\psi}) + \int_0^t (a_{42} sign^{\alpha_1}(\psi_d - \psi) + a_{41} sign^{\alpha_2}(\dot{\psi}_d - \dot{\psi}))dt
\end{aligned} \tag{12}
$$

where $\alpha_1 \in (0,1)$, $\alpha_2 = \frac{2\alpha_1}{\alpha_1+1}$, and $a_{i,j}(i,j = 1,2,3,4)$ are the coefficients of sliding surfaces that are described later. $sign^{\rho}$ is the representation of $sign^{\rho}(x) = |x|^{\rho} sign(x)$ where $\rho > 0$.

3.1. Coefficients of Sliding Surfaces

The coefficients of the sliding manifolds in (12) are derived using the same Hurwitz stability criterion as in [45,51], and they are provided as follows:

$$
\begin{aligned}
& a_{11} > 0, a_{12} > 0, a_{41} > 0, a_{42} > 0 \\
& a_{21} = -\frac{m}{u_1 \cos\psi}(r_{11}r_{12} + r_{21}r_{31} + r_{32}r_{11}) \\
& a_{22} = -\frac{m}{u_1 \cos\psi}(r_{11}r_{21}r_{31}) \\
& a_{23} = 1, a_{24} = r_{12} + r_{21} + r_{31} \\
& a_{31} = \frac{m}{u_1 \cos\phi\cos\psi}(r_{12}r_{22} + r_{22}r_{32} + r_{32}r_{12}) \\
& a_{32} = \frac{m}{u_1 \cos\phi\cos\psi}(r_{12}r_{22}r_{32}) \\
& a_{33} = 1, a_{34} = r_{12} + r_{22} + r_{32}
\end{aligned} \tag{13}
$$

where $r_{ij}(i,j = 1,2,3,4)$ are the design parameters that are constant.

3.2. Ftnsmc Controller Design

In real-world systems, neural networks (NNs) are always used as one of the online estimation methodologies for dealing with unknown nonlinear uncertainties. The nonlinear quadrotor dynamics in (10), including parametric uncertainties and external disturbances, are considered to be unknown in this section. Unknown nonlinear dynamics $f_i \ \forall i = 1,2,3,4,5,6$ are estimated as follows:

$$
f_i = \Theta_i^T \sigma\left(H_i^T \varpi_i\right) + \chi_i \tag{14}
$$

where $\Theta_i \in \Re^{2\times h}$ are the required limited NN weights meeting $\frac{k_n}{2}\sum_{i=1}^{4} \|\Theta_i\|^2 \leq \Theta_M$ and Θ_M is a positive constant, $k_n > 0$ represents the positive NN learning rate that is specified

later, $h = 8$ is the number of neurons in the hidden layer, $\sigma(H_i\varpi_i)$ represents the basis function, and $H_i^T \in \Re^{h\times n}$ represents the mapping between the inputs and the hidden-layer neurons, where $n = 7$ is the number of inputs to the NN, $\frac{1}{2}\sum_{i=1}^{4} a_{i1}\chi_i^2 \leq \chi_M$ with χ_M being a positive constant, and χ_i being the limited NN reconstruction error fulfilling.

$\hat{\Theta}_i$ represents the unknown NN weights, and the approximated uncertain dynamics are given by:

$$\hat{f}_i = \hat{\Theta}_i^T \sigma\left(H_i^T \varpi_i\right) \tag{15}$$

The NN weight estimation error is defined as $\tilde{\Theta}_i = \Theta_i - \hat{\Theta}_i$, and the estimation error dynamics are $\dot{\tilde{\Theta}}_i = -\dot{\hat{\Theta}}_i$. The error in dynamics estimation can, therefore, be described as follows:

$$\begin{aligned} \tilde{f}_i &= f_i - \hat{f}_i \\ &= \Theta_i^T \sigma\left(H_i^T \varpi_i\right) + \chi_i - \hat{\Theta}_i^T \sigma\left(H_i^T \varpi_i\right) \\ &= \tilde{\Theta}_i^T \sigma\left(H_i^T \varpi_i\right) + \chi_i \end{aligned} \tag{16}$$

The estimated NN weights were adjusted using the adaption law shown below:

$$\dot{\hat{\Theta}}_i = -k_n \hat{\Theta}_i + s_i a_{i1} \sigma\left(H_i \overline{W_i}\right), \quad \forall i = 1, 2, 3, 4. \tag{17}$$

where $k_n > 0$ is the learning rate.

An exponential reaching law is provided to achieve appropriate steady-state behavior [52]:

$$\dot{s}_i = -\eta_1 s_i - \eta_2 sign^{\beta}(s_i) \tag{18}$$

where $\eta_1 > 0$,$\eta_2 > 0$ and $\beta \in (0, 1)$.

Taking the derivative of sliding surface s_i defined in (12) yields:

$$\begin{aligned} \dot{s}_1 &= a_{11}(\ddot{z}_d - \ddot{z}) + a_{12} sign^{\alpha_1}(z_d - z) + a_{11} sign^{\alpha_2}(\dot{z}_d - \dot{z}) \\ \dot{s}_2 &= a_{21}(\ddot{x}_d - \ddot{x}) + a_{22} sign^{\alpha_1}(x_d - x) + a_{21} sign^{\alpha_2}(\dot{x}_d - \dot{x}) \\ &\quad + a_{23}(\ddot{\phi}_d - \ddot{\phi}) + a_{24} sign^{\alpha_1}(\phi_d - \phi) + a_{23} sign^{\alpha_2}(\dot{\phi}_d - \dot{\phi}) \\ \dot{s}_3 &= a_{31}(\ddot{y}_d - \ddot{y}) + a_{32} sign^{\alpha_1}(y_d - y) + a_{31} sign^{\alpha_2}(\dot{y}_d - \dot{y}) \\ &\quad + a_{33}\left(\ddot{\theta}_d - \ddot{\theta}\right) + a_{34} sign^{\alpha_1}(\theta_d - \theta) + a_{33} sign^{\alpha_2}\left(\dot{\theta}_d - \dot{\theta}\right) \\ \dot{s}_4 &= a_{41}(\ddot{\psi}_d - \ddot{\psi}) + a_{42} sign^{\alpha_1}(\psi_d - \psi) + a_{41} sign^{\alpha_2}(\dot{\psi}_d - \dot{\psi}) \end{aligned} \tag{19}$$

The dynamics of the system, including parametric uncertainties and external disturbances, were considered to be unknown and they were estimated with the NN in this work. Therefore, by inserting the estimation function in (16) for all the dynamics, and taking the reaching law into consideration in (18) for $\dot{s}_i = 0$ control signals can be written as follows:

$$\begin{aligned} u_1 &= \frac{1}{a_{11}}\left\{a_{11}\left(\ddot{z}_d - \hat{f}_1\right) + a_{12}\, sign^{\alpha_1}(z_d - z) + a_{11}\, sign^{\alpha_2}(\dot{z}_d - \dot{z}) + \eta_1 s_1 + \eta_2 sign^{\beta}(s_1)\right\} \\ u_2 &= \frac{1}{a_{21} + a_{23}}\left\{\begin{array}{l} a_{21}\left(\ddot{x}_d - \hat{f}_2\right) + a_{22} sign^{\alpha_1}(x_d - x) + a_{21} sign^{\alpha_2}(\dot{x}_d - \dot{x}) \\ + a_{23}\left(\ddot{\phi}_d - \hat{f}_3\right) + a_{24} sign^{\alpha_1}(\phi_d - \phi) + a_{23} sign^{\alpha_2}(\dot{\phi}_d - \dot{\phi}) + \eta_1 s_2 + \eta_2 sign^{\beta}(s_2) \end{array}\right\} \\ u_3 &= \frac{1}{a_{31} + a_{33}}\left\{\begin{array}{l} a_{31}\left(\ddot{y}_d - \hat{f}_4\right) + a_{32} sign^{\alpha_1}(y_d - y) + a_{31} sign^{\alpha_2}(\dot{y}_d - \dot{y}) + a_{33}\left(\ddot{\theta}_d - \hat{f}_5\right) \\ + a_{34} sign^{\alpha_1}(\theta_d - \theta) + a_{33} sign^{\alpha_2}\left(\dot{\theta}_d - \dot{\theta}\right) + \eta_1 s_3 + \eta_2 sign^{\beta}(s_3) \end{array}\right\} \\ u_4 &= \frac{1}{a_{41}}\left\{a_{41}\left(\ddot{\psi}_d - \hat{f}_1\right) + a_{42} sign^{\alpha_1}(\psi_d - \psi) + a_{41} sign^{\alpha_2}(\dot{\psi}_d - \dot{\psi}) + \eta_1 s_4 + \eta_2 sign^{\beta}(s_4)\right\} \end{aligned} \tag{20}$$

3.3. Stability Analysis

Theorem 1. *Consider nonlinear System (10) with the sliding surface dynamics in (19) , the controller inputs as in (20), and the estimated dynamics applied to the system, and the estimated NN weights tuned with the adaptation law (17), All NN weight estimation errors and sliding surfaces were SGPFS, and the tracking error converged in a finite time to a small neighborhood of the origin.*

Proof of Theorem 1. Define the Lyapunov candidate function of all sliding surfaces, s_i and NN weight estimation errors $\tilde{\Theta}_i$ as follows:

$$V = \sum_{i=1}^{4} \left(\frac{1}{2} {s_i}^2 + \frac{1}{2} \left(\tilde{\Theta}_i^T \tilde{\Theta}_i \right) \right) \tag{21}$$

by taking Derivative (21) and inserting the dynamics in (10), and considering that $\dot{\tilde{\Theta}}_i = -\dot{\hat{\Theta}}_i$ to obtain:

$$\begin{aligned}
\dot{V} &= \sum_{i=1}^{4} \left(s_i \dot{s}_i + \tilde{\Theta}_i^T \dot{\tilde{\Theta}}_i \right) \\
&= s_1 \dot{s}_1 + s_2 \dot{s}_2 + s_3 \dot{s}_3 + s_4 \dot{s}_4 + \tilde{\Theta}_1^T \dot{\tilde{\Theta}}_1 + \tilde{\Theta}_2^T \dot{\tilde{\Theta}}_2 + \tilde{\Theta}_3^T \dot{\tilde{\Theta}}_3 + \tilde{\Theta}_4^T \dot{\tilde{\Theta}}_4 \\
&= s_1 \left(a_{11} (\ddot{z}_d - f_1 + g_1 u_1 + \Delta f_1) + a_{12} sign^{\alpha_1} (z_d - z) + a_{11} sign^{\alpha_2} (\dot{z}_d - \dot{z}) \right) \\
&+ s_2 \begin{pmatrix} a_{21} (\ddot{x}_d - f_2 + g_2 u_2 + \Delta f_2) + a_{22} sign^{\alpha_1} (x_d - x) + a_{21} sign^{\alpha_2} (\dot{x}_d - \dot{x}) \\ + a_{23} (\ddot{\phi}_d - f_3 + g_3 u_2 + \Delta f_3) + a_{24} sign^{\alpha_1} (\phi_d - \phi) + a_{23} sign^{\alpha_2} (\dot{\phi}_d - \dot{\phi}) \end{pmatrix} \\
&+ s_3 \begin{pmatrix} a_{31} (\ddot{y}_d - f_4 + g_4 u_3 + \Delta f_4) + a_{32} sign^{\alpha_1} (y_d - y) + a_{31} sign^{\alpha_2} (\dot{y}_d - \dot{y}) \\ + a_{33} (\ddot{\theta}_d - f_5 + g_5 u_3 + \Delta f_5) + a_{34} sign^{\alpha_1} (\theta_d - \theta) + a_{33} sign^{\alpha_2} (\dot{\theta}_d - \dot{\theta}) \end{pmatrix} \\
&+ s_4 \left(a_{41} (\ddot{\psi}_d - f_6 + g_6 u_4 + \Delta f_6) + a_{42} sign^{\alpha_1} (\psi_d - \psi) + a_{41} sign^{\alpha_2} (\dot{\psi}_d - \dot{\psi}) \right) \\
&- \tilde{\Theta}_1^T \dot{\hat{\Theta}}_1 - \tilde{\Theta}_2^T \dot{\hat{\Theta}}_2 - \tilde{\Theta}_3^T \dot{\hat{\Theta}}_3 - \tilde{\Theta}_4^T \dot{\hat{\Theta}}_4
\end{aligned} \tag{22}$$

Using the controller inputs in (20), dynamic estimation error function in (16) and reaching law in (18), (22) can be rewritten in a compact form as:

$$\begin{aligned}
\dot{V} &= -\sum_{i=1}^{4} \eta_1 s_i^2 - \sum_{i=1}^{4} \eta_2 |s_i|^{\beta+1} + \sum_{i=1}^{4} s_i a_{i1} \tilde{f}_1 - \sum_{i=1}^{4} \tilde{\Theta}_i^T \dot{\hat{\Theta}}_i \\
&= -\sum_{i=1}^{4} \eta_1 s_i^2 - \sum_{i=1}^{4} \eta_2 |s_i|^{\beta+1} + \sum_{i=1}^{4} s_i a_{i1} \left(\tilde{\Theta}_i^T \sigma \left(H_i^T \varpi_i \right) + \chi_i \right) - \sum_{i=1}^{4} \tilde{\Theta}_i^T \dot{\hat{\Theta}}_i \\
&= -\sum_{i=1}^{4} \eta_1 s_i^2 - \sum_{i=1}^{4} \eta_2 |s_i|^{\beta+1} + \sum_{i=1}^{4} s_i a_{i1} \left(\tilde{\Theta}_i^T \sigma \left(H_i^T \varpi_i \right) \right) + \sum_{i=1}^{4} s_i a_{i1} \chi_i - \sum_{i=1}^{4} \tilde{\Theta}_i^T \left(-k_n \hat{\Theta}_i + s_i a_{i1} \sigma \left(H_i \overline{W}_i \right) \right) \\
&= -\sum_{i=1}^{4} \eta_1 s_i^2 - \sum_{i=1}^{4} \eta_2 |s_i|^{\beta+1} + \sum_{i=1}^{4} s_i a_{i1} \left(\tilde{\Theta}_i^T \sigma \left(H_i^T \varpi_i \right) \right) + \sum_{i=1}^{4} s_i a_{i1} \chi_i + k_n \sum_{i=1}^{4} \tilde{\Theta}_i^T \hat{\Theta}_i - \sum_{i=1}^{4} s_i a_{i1} \tilde{\Theta}_i^T \left(\sigma \left(H_i \overline{W}_i \right) \right) \\
&= -\sum_{i=1}^{4} \eta_1 s_i^2 - \sum_{i=1}^{4} \eta_2 |s_i|^{\beta+1} + \sum_{i=1}^{4} s_i a_{i1} \chi_i + k_n \sum_{i=1}^{4} \tilde{\Theta}_i^T \left(\Theta_i - \tilde{\Theta}_i \right) \\
&= -\sum_{i=1}^{4} \eta_1 s_i^2 - \sum_{i=1}^{4} \eta_2 |s_i|^{\beta+1} + \sum_{i=1}^{4} s_i a_{i1} \chi_i - k_n \sum_{i=1}^{4} \left\| \tilde{\Theta}_i \right\|^2 + k_n \sum_{i=1}^{4} \tilde{\Theta}_i^T \Theta_i
\end{aligned} \tag{23}$$

The following inequalities hold from Young's inequality in Lemma 3:

$$k_n \sum_{i=1}^{4} \tilde{\Theta}_i^T \Theta_i \leq \frac{k_n}{2} \sum_{i=1}^{4} \left\| \tilde{\Theta}_i \right\|^2 + \frac{k_n}{2} \sum_{i=1}^{4} \| \Theta_i \|^2$$
$$\sum_{i=1}^{4} s_i a_{i1} \chi_i \leq \frac{1}{2} \sum_{i=1}^{4} s_i^2 a_{i1} + \frac{1}{2} \sum_{i=1}^{4} a_{i1} \chi_i^2 \tag{24}$$

Upper bounds Θ_M and χ_M; inserting (24) into (23), we obtain:

$$\begin{aligned} \dot{V} &\leq -\sum_{i=1}^{4} \eta_1 s_i^2 - \sum_{i=1}^{4} \eta_2 |s_i|^{\beta+1} + \frac{1}{2} \sum_{i=1}^{4} s_i^2 a_{i1} + \frac{1}{2} \sum_{i=1}^{4} a_{i1} \chi_i^2 - k_n \sum_{i=1}^{4} \left\| \tilde{\Theta}_i \right\|^2 + \frac{k_n}{2} \sum_{i=1}^{4} \left\| \tilde{\Theta}_i \right\|^2 + \frac{k_n}{2} \sum_{i=1}^{4} \| \Theta_i \|^2 \\ &\leq -\sum_{i=1}^{4} \left(\eta_1 - \frac{a_{i1}}{2} \right) s_i^2 - \sum_{i=1}^{4} \eta_2 |s_i|^{\beta+1} - \frac{k_n}{2} \sum_{i=1}^{4} \left\| \tilde{\Theta}_i \right\|^2 + 4\chi_M + 4\Theta_M \end{aligned} \tag{25}$$

Here, by using Lemma 2, the inequality below holds:

$$\left(\sum_{i=1}^{4} \eta_2 |s_i|^2 \right)^{\frac{\beta+1}{2}} \leq \sum_{i=1}^{4} \eta_2 |s_i|^{\beta+1} \tag{26}$$

Moreover, according to Lemma 4, the inequalities. below exist:

$$\left(\left\| \tilde{\Theta}_i \right\|^2 \right)^{\xi} \leq (1-\xi) \xi^{\frac{\xi}{1-\xi}} + \left\| \tilde{\Theta}_i \right\|^2$$
$$\left(\|s\|^2 \right)^{\xi} \leq (1-\xi) \xi^{\frac{\xi}{1-\xi}} + \|s\|^2 \tag{27}$$

where $0 < \xi = \frac{\beta+1}{2} < 1$.

By using (26) and (27), and Lemma 2, (25) can be rewritten as:

$$\begin{aligned} \dot{V} &\leq -\left(\eta_1 + \eta_2 - \frac{a_{i1}}{2} \right) \left(\sum_{i=1}^{4} \frac{1}{2} |s_i|^2 \right)^{\xi} - k_n \left(\sum_{i=1}^{4} \frac{1}{2} \left\| \tilde{\Theta}_i \right\|^2 \right)^{\xi} + 4\chi_M + 4\Theta_M + \left(\frac{k_n}{2} + \eta_1 - \frac{a_{i1}}{2} \right) \left((1-\xi) \xi^{\frac{\xi}{1-\xi}} \right) \\ &\leq -\bar{c} V_i^{\xi} + \bar{\delta} \end{aligned} \tag{28}$$

where $\bar{c} = \min\left\{ -\left(\eta_1 + \eta_2 - \frac{a_{i1}}{2} \right), k_n \right\}$,
$\bar{\delta} = 4\chi_M + 4\Theta_M + \left(\frac{k_n}{2} + \eta_1 - \frac{a_{i1}}{2} \right) \left((1-\xi) \xi^{\frac{\xi}{1-\xi}} \right)$, with $\eta_1 \geq \frac{a_{i1}}{2}$.

Now let

$$\bar{T} = \frac{1}{(1-\xi)\gamma\bar{c}} \left(V^{1-\xi}(s_i(0), \tilde{\Theta}_i(0)) - \left(\frac{\bar{\delta}}{(1-\gamma)\bar{c}} \right)^{\frac{1-\xi}{\xi}} \right) \tag{29}$$

With Lemma 1, all signals are SGPFS, $\forall t \geq \bar{T}$, $V^{\xi}(s_i, \tilde{\Theta}_i) \leq \frac{\bar{\delta}}{(1-\xi)\bar{c}}$.

Moreover, from the definition of V, $\forall t \geq \bar{T}$, the following can be obtained:

$$|e_i| \leq \left(\frac{\bar{\delta}}{(1-\xi)\bar{c}} \right)^{\frac{1}{2\xi}} \tag{30}$$

After a finite time $\bar{T}$, the tracking errors of all states e_i converge to a small neighborhood of the origin and stay there. This concludes the proof. □

Remark 1. *According to stability analysis, design parameters $\xi, \gamma, \bar{c}$, and $\bar{\delta}$ significantly impacted the proposed controller's capacity for stability. High control precision and a quick convergence rate can result from large $\xi, \gamma, \bar{c}$. They could, however, provide comparatively huge control torques that would not be practical. High control precision and a quick convergence rate can also come from large $\bar{\delta}$. This might, however, potentially cause an infinite convergence. Therefore, it is important to carefully choose the design parameters of the suggested controller so that it can reach a sufficient level of control performance.*

Remark 2. *Due to their fractional power structure, finite-time-based controllers are more reliable than traditional controllers, as is demonstrated in the simulation results in the next section. However, because initial conditions impact the convergence time of the finite-time controller, this is not always possible in applications. Therefore, it is not always possible to compute the settling time of the finite-time controller. Since the estimated function's initial values are included in this study, it is also not possible to accurately compute the settling time. A fixed-time stability structure was suggested in the literature [53] as a solution to this issue, and this structure may serve as the subject of our next research.*

Next, numerical simulations are given in detail.

4. Numerical Simulations

To test the efficiency of the suggested controller, a nonlinear model of a quadrotor UAV was built and simulations in MATLAB were run; ode45 was used as the solver. Two different trajectory tracking simulations were performed on quadrotor UAV transporting a suspended payload system to demonstrate the effectiveness of the developed controller. Furthermore, the simulations were enlarged with two more control approaches, demonstrating the superiority of the suggested control over other control structures. First, a classical SMC was used to control the system. Then, the system was controlled with a finite-time SMC (FTSMC) control structure by adding a finite-time control structure to the classical SMC structure. Thus, before adding the neural-network component, the advantages of the finite-time control structure over the classical SMC were observed. Lastly, we clearly demonstrate the superiority of the results obtained using the proposed controller, finite-time-based neuro-sliding mode controller (FTNSMC) over the results obtained in the previous simulations.

The parameters of the quadrotor dynamical model are given in Table 1. The UAV's mass and inertia were also subject to a %20 uncertainty as $\Delta m = 0.20m$, $\Delta I_x = 0.20I_x$, $\Delta I_y = 0.20I_y$, $\Delta I_z = 0.20I_z$. Moreover, the quadrotor was assumed to be subjected to constant time-varying disturbances as $D_x = 0.02sin(t+5)$, $D_y = 0.02cos(2t+3)$, $D_z = 0.02sin(3t+2)$, $D_z = 0.02sin(3t+2)$, $D_\theta = 0.02sin(0.2t+5)$, $D_\phi = 0.02cos(0.2t+3)$, $D_\psi = 0.02sin(0.3t+2)$. Moreover, payload's desired angles are chosen as, $\alpha_x = A_{\alpha_x}cost$, $\alpha_y = A_{\alpha_y}sint$, where $A_{\alpha_x} = A_{\alpha_y} = 0.5$. Controller parameters were selected by trial and error as in Table 2. The hidden-layer neurons were employed at random, while the neural-network weights were initially set to zero during the simulations. The simulations were run with the following assumptions in mind.

Assumption 1. *The parametric uncertainties and external disturbances that were defined in (9) were assumed to be bound by a known bound.*

$$sat(x) = \begin{cases} x/\kappa, & |x| \leq \kappa \\ sign(x), & |x| > \kappa \end{cases} \tag{31}$$

where $\kappa > 0$, and it was taken to be $\kappa = 0.5$ in the simulations.

Assumption 2. *The parametric uncertainties and external disturbances that were defined in (9) were assumed to be bounded by a known bound.*

Table 1. Model parameters.

Parameter	Value	Parameter	Value
m	1.1 kg	g	9.8 m/s^2
m_L	0.5 kg	L	1 m
I_x	1.22 Ns^2/rad	l	0.21 m
I_y	1.22 Ns^2/rad	b	5 Ns
I_z	2.22 Ns^2/rad	d	2 N/ms^2
K_1	0.1 Ns/m	K_4	0.12 Ns/rad
K_2	0.1 Ns/m	K_5	0.12 Ns/rad
K_3	0.1 Ns/m	K_6	0.12 Ns/rad
J_r	0.2 Ns^2/rad		

Table 2. Controller parameters.

Parameter	Value	Parameter	Value
a_{11}	0.01	r_{22}	0.3
a_{12}	0.05	r_{32}	0.03
a_{41}	0.01	η_1	2
a_{42}	0.02	η_2	5
r_{11}	0.03	β	1/3
r_{12}	0.3	α_1	83/97
r_{31}	0.03	k_n	0.1
r_{21}	0.03		

Two different scenarios were considered in the simulations for the trajectory tracking problem of quadrotor carrying a suspended payload.

4.1. Scenario 1

In the context of model uncertainties and external disturbances, the control goal was to allow the quadrotor to track a square desired trajectory. For simulation experiments, the quadrotor's beginning position and angle parameters were $[0, 0, 0]$m and $[0, 0, 0]$rad, and the chosen desired quadrotor trajectories are listed in Table 3 . The simulation time for all runs was 80 s.

Table 3. Desired trajectories for the first scenario.

Parameter	Trajectories	Time (s)
$[x_d \ \ y_d \ \ z_d]$	$[6, 6, 6]$ m	10
	$[3, 6, 6]$ m	20
	$[3, 3, 6]$ m	30
	$[6, 3, 6]$ m	40
	$[6, 6, 6]$ m	50
	$[6, 6, 0]$ m	80
$[\phi \ \ \theta_d \ \ \psi_d]$	$[0, 0, 0.5]$ rad	60
	$[0, 0, 0]$ rad	80

The x position of the quadrotor for three controllers is given in Figure 4. For a classical SMC, the quadrotor reached the first desired x value at around 7.89 s. The addition of a finite-time structure decreased the reaching time to around 6.08 s, and with the proposed method, the quadrotor reached the first desired x value at around 6.16 s.

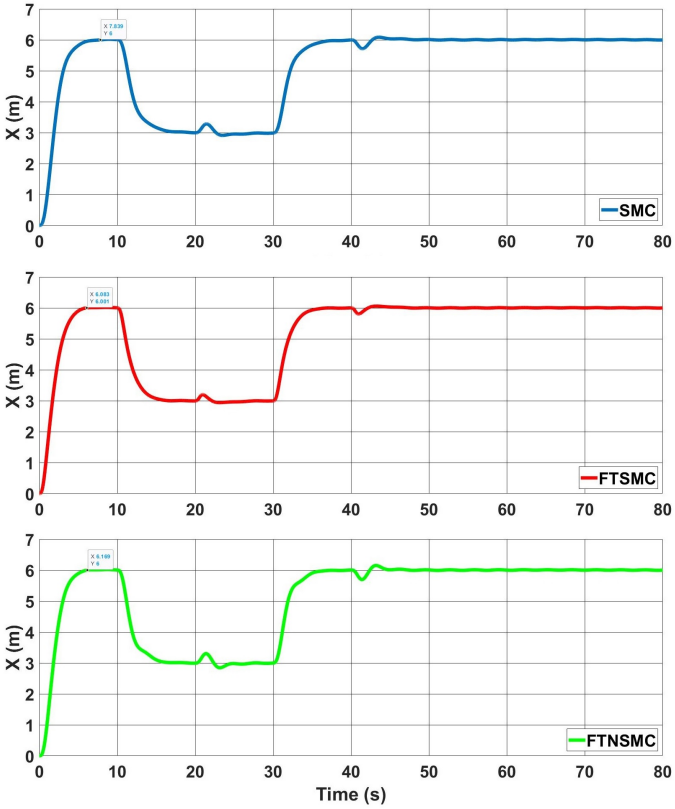

Figure 4. x positions of quadrotor for Scenario 1.

The y position of the quadrotor for three controllers is given in Figure 5. For a classical SMC, the quadrotor could not reach the first desired y value but it reaches about 5.97 m at 10 s. With FTSMC, the quadrotor reached the first desired y value at around 8.87 s, and with the addition of the neural-network component, this value decreased to around 8.55 s.

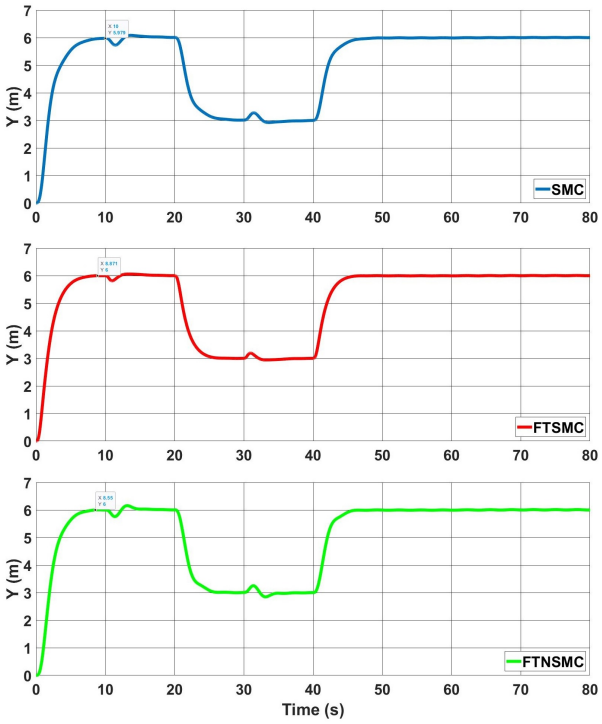

Figure 5. y positions of quadrotor for Scenario 1.

The z position of the quadrotor for three controllers is given in Figure 6. The quadrotor arrived at the desired 6 m value at around 12.68 s, whereas for FTSMC, this value decreased to 10.53 s; with the suggested controller, it settled to the desired value in less than 8.46 s.

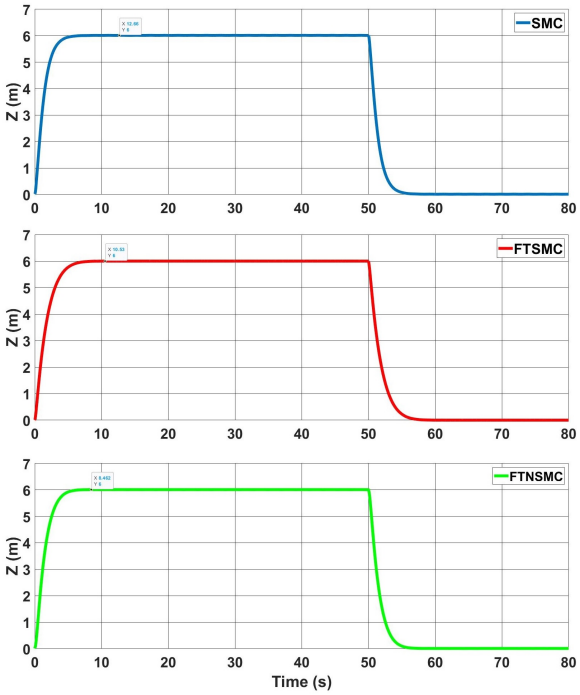

Figure 6. z positions of quadrotor for Scenario 1.

In Figure 7, the change in controller input u_1 is given. Although the initial values of the control input were similar in all three controllers, the chattering effect was clearly observed for the classical SMC. With FTSMC, this chattering decreased but remained. Using the proposed controller, the chattering effect completely disappeared. Moreover, the proposed controller exceptionally decreased the input value compared to SMC and FTSMC.

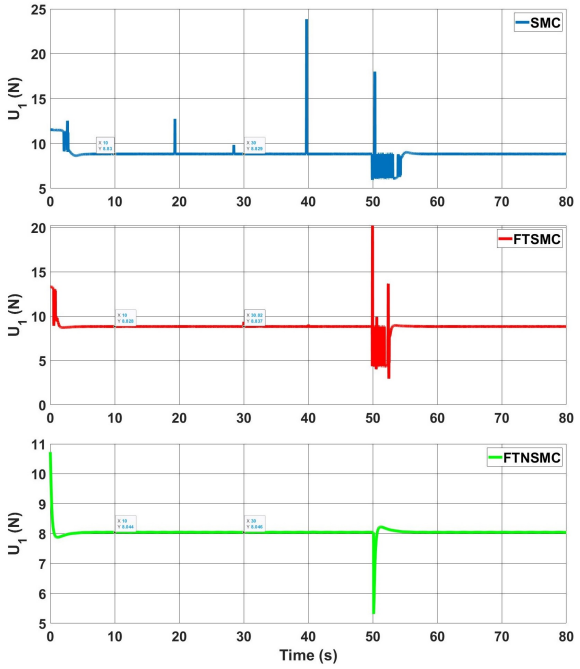

Figure 7. u_1 control inputs for Scenario 1.

Figure 8 displays the change in sliding variable s_1. It behaved as expected when the variables approached its sliding surfaces. Additionally, the oscillations of the sliding variable completely accounted for the position and velocity tracking inaccuracies of the system state variables.

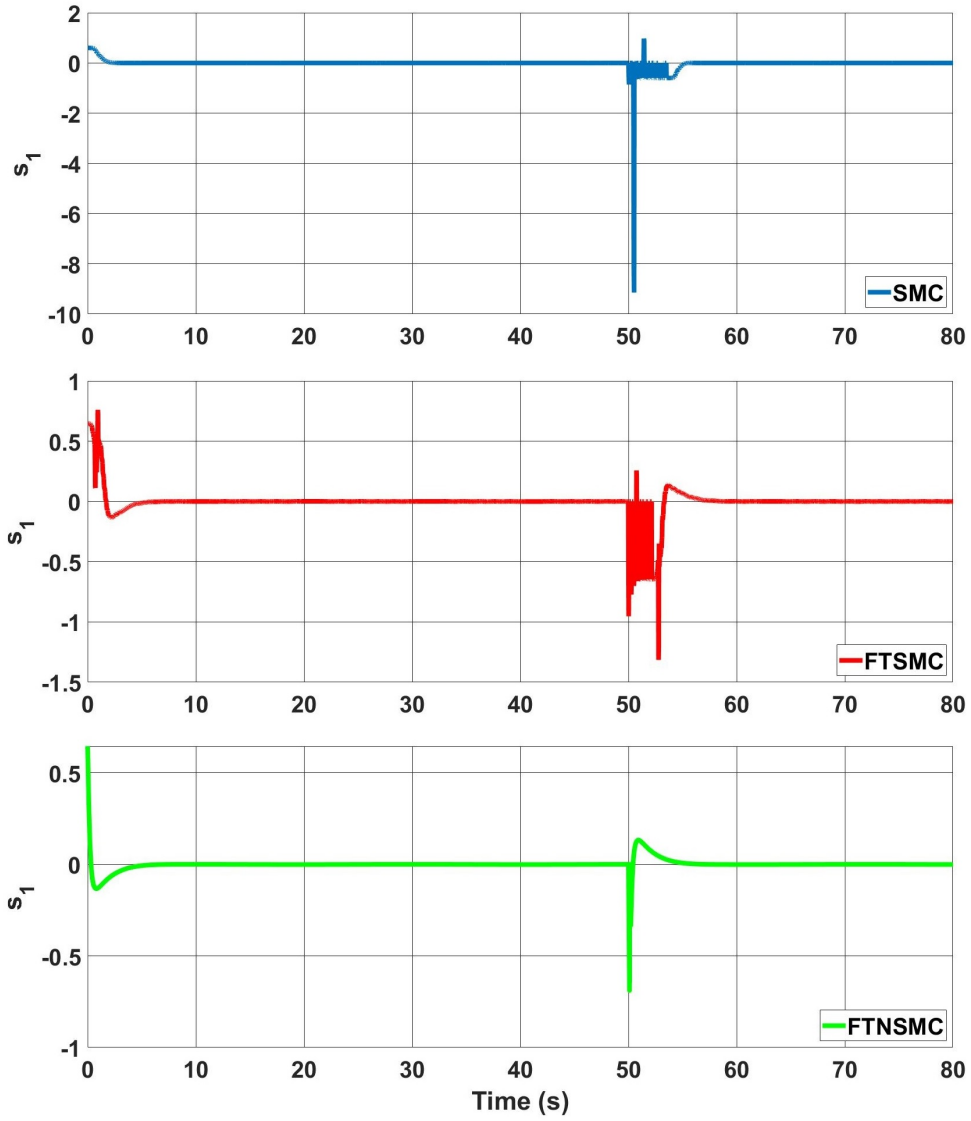

Figure 8. s_1 sliding variables for Scenario 1.

Variation in controller inputs u_2 and u_3 is seen in Figures 9 and 10 respectively. When SMC was used, both a chattering effect and a very high control input value were observed. By adding finite time to SMC, the chattering decreased as the value of the control input decreased with FTSMC. However, in both cases, high sparks and chattering could be seen clearly. Using the proposed controller not only significantly reduced the value of the control input, but also completely eliminated the chattering effect.

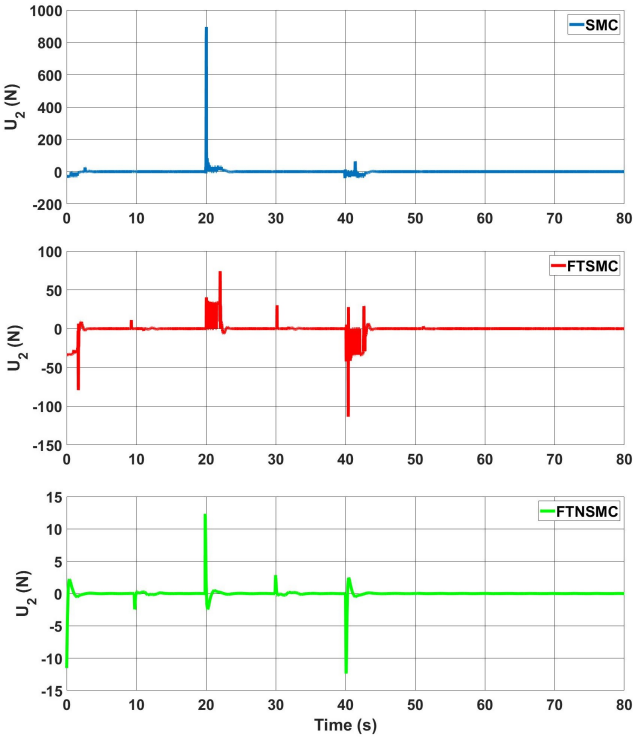

Figure 9. u_2 control inputs for Scenario 1.

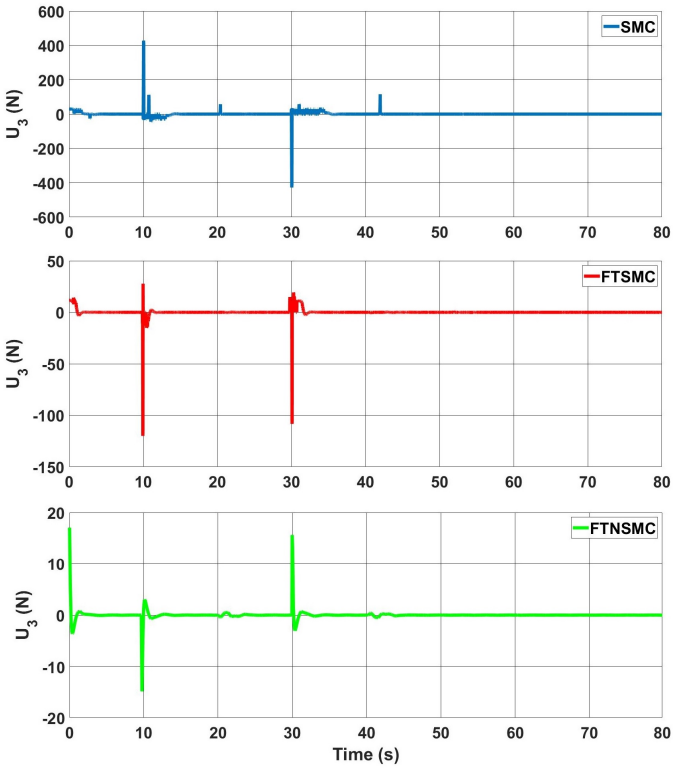

Figure 10. u_3 control inputs for Scenario 1.

In Figure 11, the change in controller input u_4 is given. Despite the fact that the starting values of the control input were comparable in all three controllers, the chattering effect was plainly seen in the conventional SMC. This chattering was reduced by FTSMC, but it still existed. When the proposed controller was used, the chattering effect was fully eliminated.

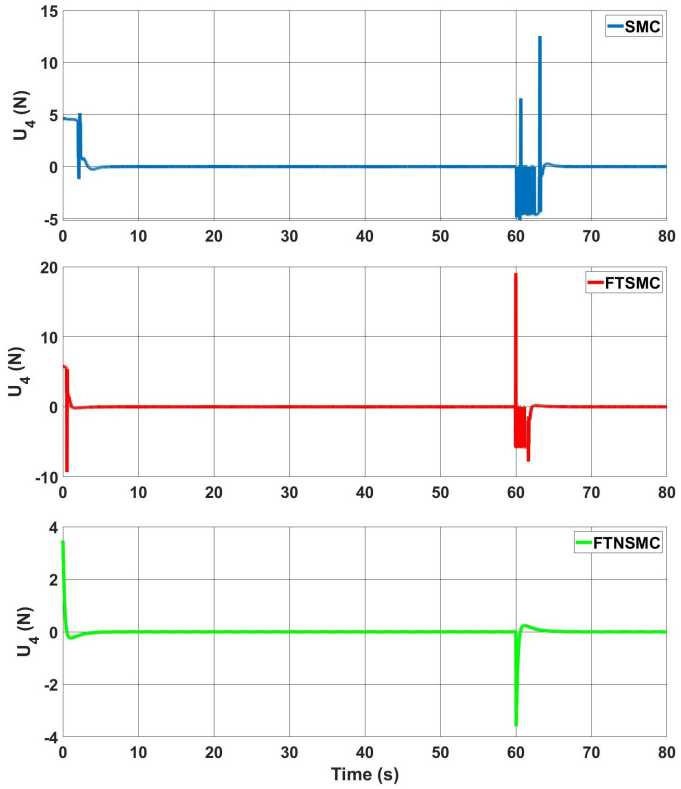

Figure 11. u_4 control inputs for Scenario 1.

Variations in quadrotor angles for proposed controller are given in Figure 12.

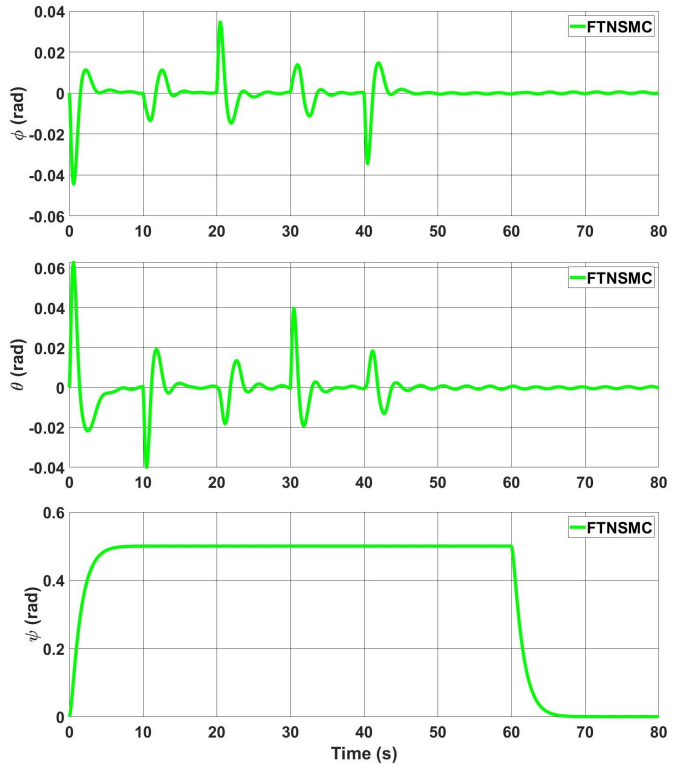

Figure 12. Quadrotor angles for proposed controller for Scenario 1.

The trajectory tracking of the UAV in 3D is given in Figure 13 for all three controllers. Adding a finite-time structure to the conventional SMC obviously sped up the system responsiveness. Additionally, the NN improved the system response even more.

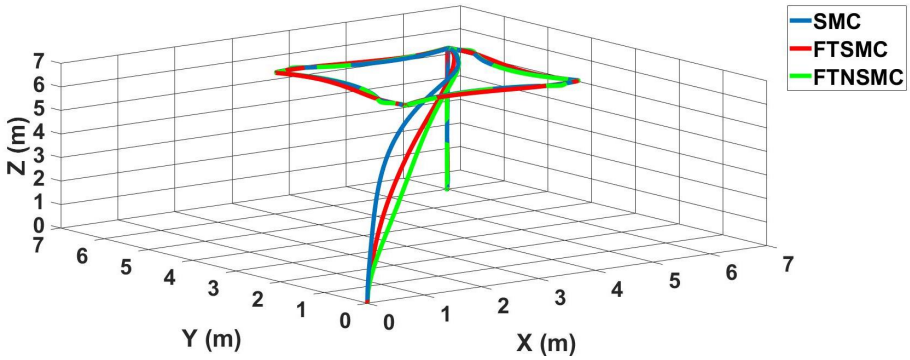

Figure 13. Trajectory of quadrotor in 3D space for Scenario 1.

Figure 14 shows the variation in the neural-network weights. As expected, the neural-network weights rapidly learnt the unknown dynamics after each movement of the UAV and converged to a value close to zero.

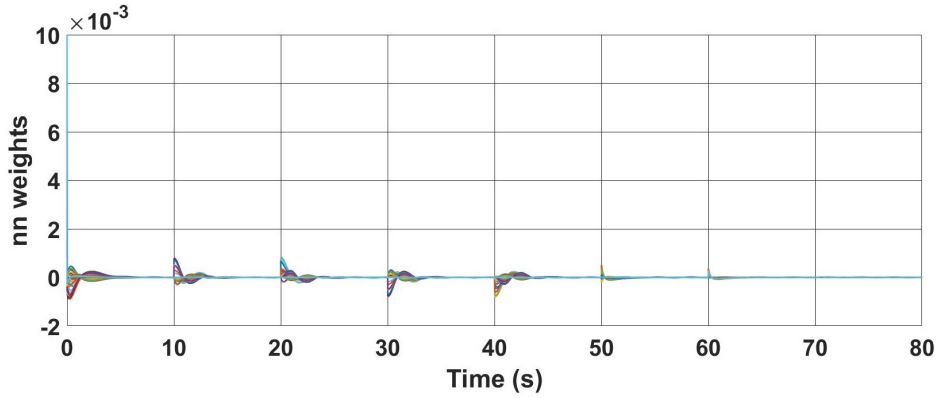

Figure 14. Change in neural-network weights for Scenario 1.

4.2. Scenario 2

In the second simulation, the control goal was to allow the quadrotor to track a circular desired trajectory and also ensures that the system state errors reach zero or to a close neighbourhood of zero in a finite time. The chosen desired quadrotor trajectories are listed in Table 4. The simulation time for all runs was 320 s.

Table 4. Desired trajectories for the second scenario.

Parameter	Trajectories
$[x_d \;\; y_d \;\; z_d]$	$50sin(0.05t)$ $50cos(0.05t)$ t
$[\theta_d \;\; \phi_d \;\; \psi_d]$	$[0, 0, 0.5]$ rad

The errors in the x axis for three controllers are given in Figure 15. Compared to a classical SMC, the proposed controller provided a fast convergence rate as accepted. The error in the x axis reached zero for the first time at around 4.3 s with the proposed controller and oscillated in the neighborhood of zero with a small margin because of the nature of desired trajectory, whereas it reached zero at around 6.6 s with FTSMC and 14 s with classical SMC.

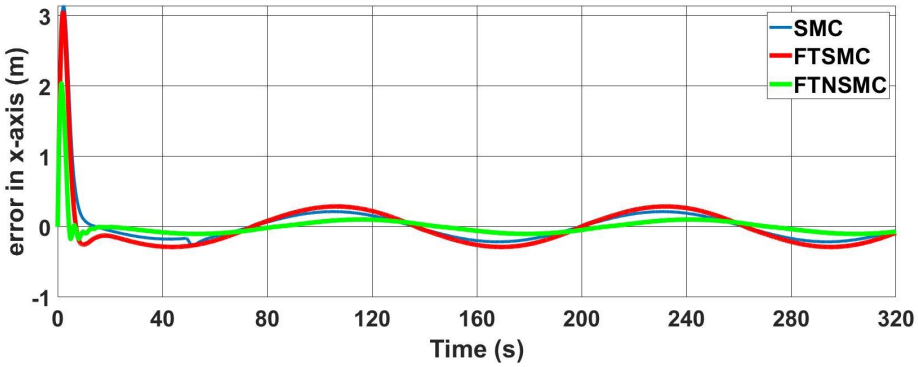

Figure 15. Errors in x-axis for Scenario 2.

Figure 16 displays the y-axis errors for three controllers. With the SMC, error convergence was very slow and it converged to zero after 50 s, as seen in the figure. The addition of finite-time structure decreased this value to around 13 s.

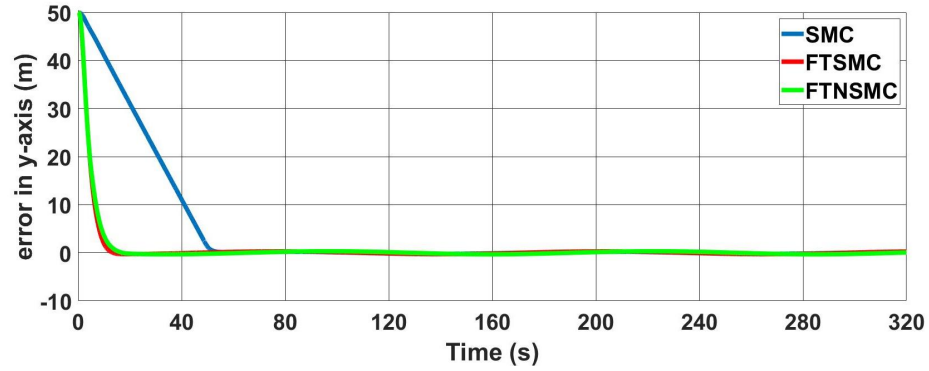

Figure 16. Errors in y-axis for Scenario 2.

In Figure 17 errors in the z axis are shown. For SMC controller error reached the neighborhood of zero within around 6.5 s. With FTSMC, this value was around 5.5 s, and with FTNSMC, it was around 4.5 s.

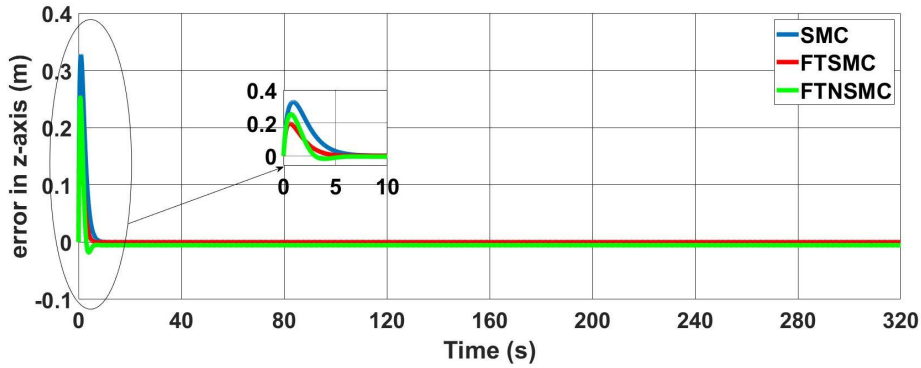

Figure 17. Errors in z-axis for Scenario 2.

u_1 controller inputs for three controllers are given in Figure 18. Since the desired trajectory was orbital, the progress of the motion in the z axis was expected to be slow. Changes in the control inputs for this axis were, therefore, close to each other within the three controllers. On the other hand, there was a difference between the convergence of the control inputs to nominal value. This time was around 4.5 s for SMC and FTSMC, and this time decreased to 2.8 s for the proposed controller. Moreover, small chattering could be detected with FTSMC, but this effect was also eliminated with FTNSMC.

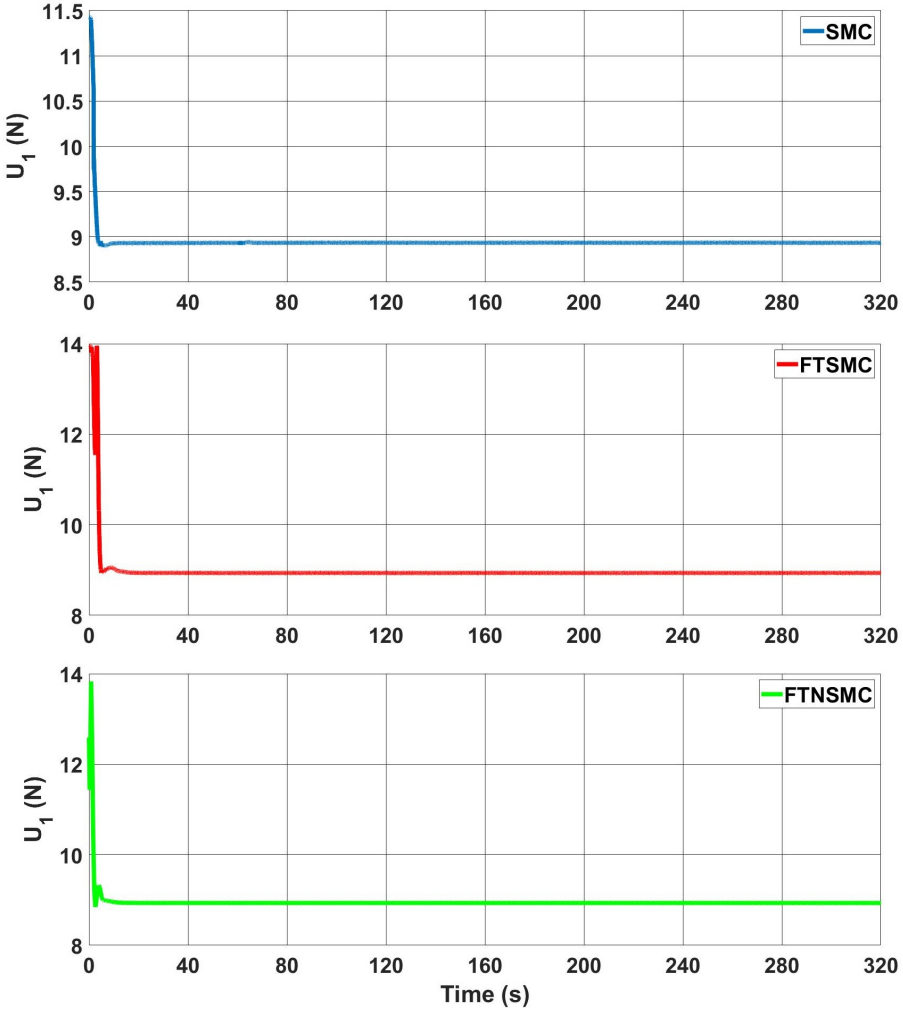

Figure 18. u_1 control inputs for Scenario 2.

Variations in control inputs u_2 and u_3 are given for the three controllers in Figures 19 and 20, respectively. The control signal for SMC exhibited undesirable sparks at the beginning with some chattering effect and, since it could only enter the desired trajectory after 50 s, high chattering occurred at these time values for input u_2. FTSMC reduced this chattering effect while increasing the initial value of control input; lastly, by using NNFTSMC, a chattering-free control input could be obtained while decreasing the initial value of control input of FTSMC. For the u_3 control input, a chattering effect could also be seen for SMC and FTSMC. However, the neural-network component in the proposed controller significantly reduced this effect.

Figure 21 demonstrates the change of control inputs u_4 for the three controllers. Although the initial values were close to each other for all controllers, with the proposed controller, the control input converged to zero more quickly after the initial movement and it was a smoother control input than the SMC and FTSMC methods. Furthermore, FTSMC showed more chattering-free control input behavior than the classical SMC did.

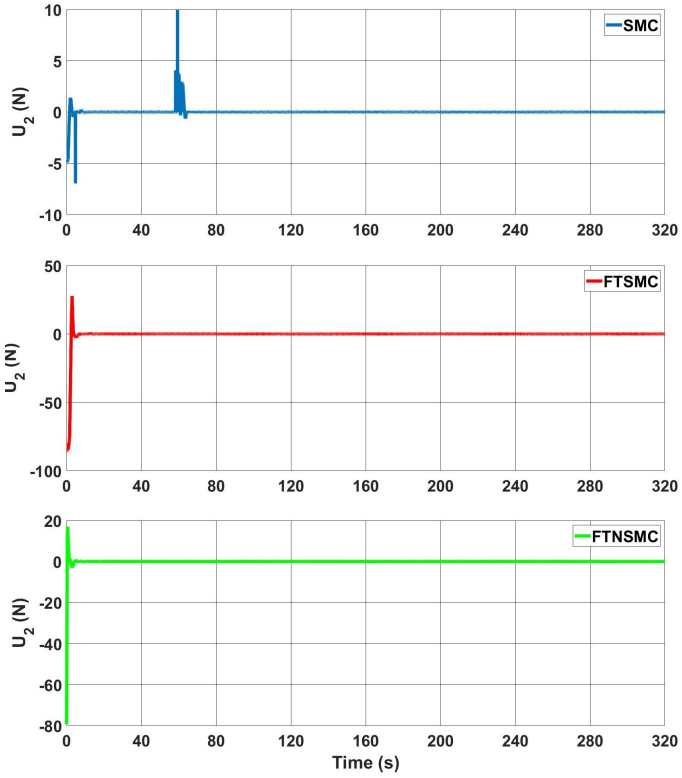

Figure 19. u_2 control inputs for Scenario 2.

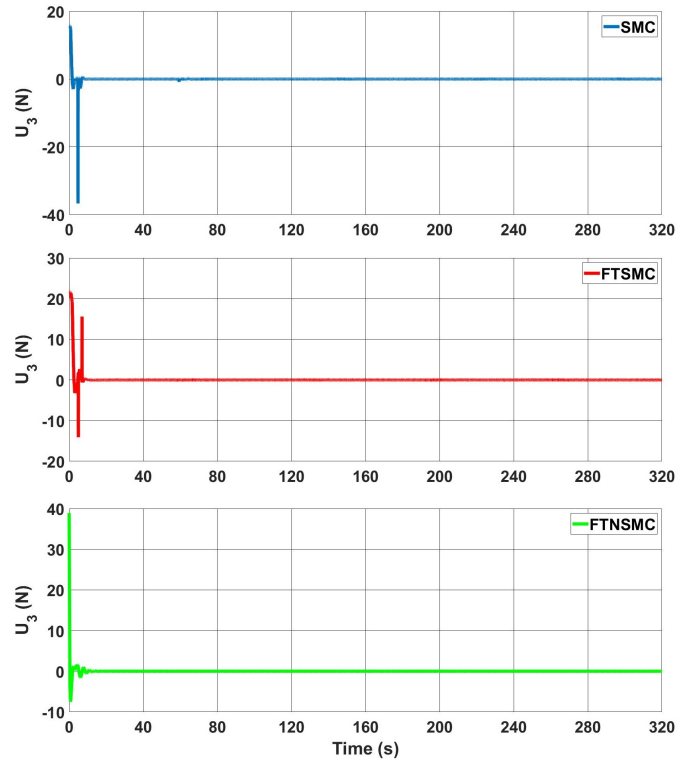

Figure 20. u_3 control inputs for Scenario 2.

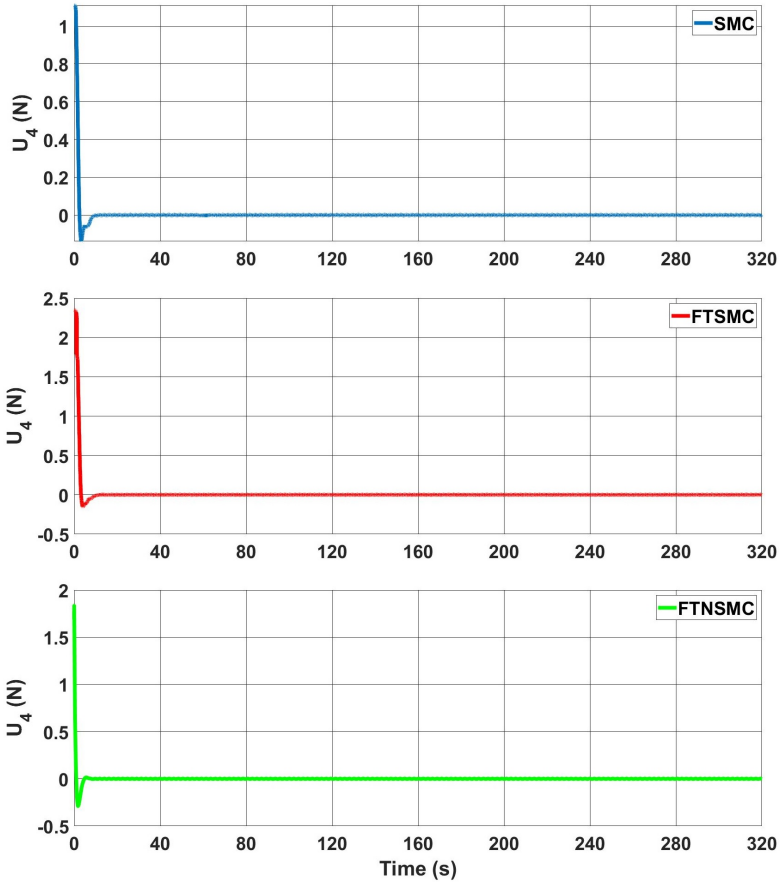

Figure 21. u_4 control inputs for Scenario 2.

In Figure 22, the trajectory of the quadrotor in 3D space is given for three controllers. The suggested controller tracking performance of the desired trajectory was superior to that of classical SMC and FTSMC.

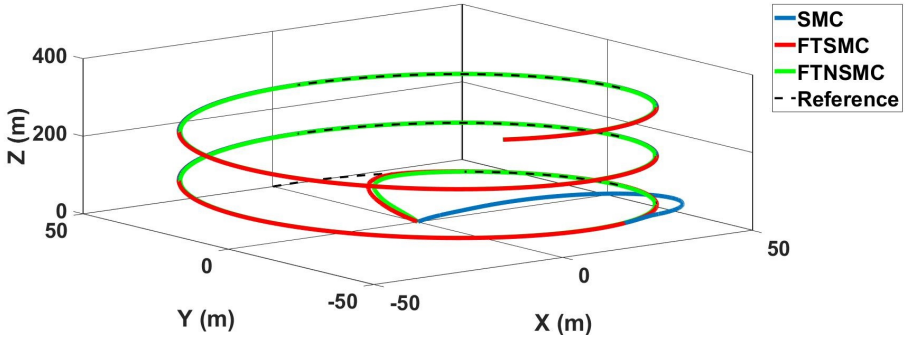

Figure 22. Trajectory of quadrotor in 3D space for Scenario 2.

The change in neural-network weights is presented in Figure 23. As expected, the neural-network weights rapidly learnt the unknown dynamics after each movement of the UAV and converged to a value close to zero.

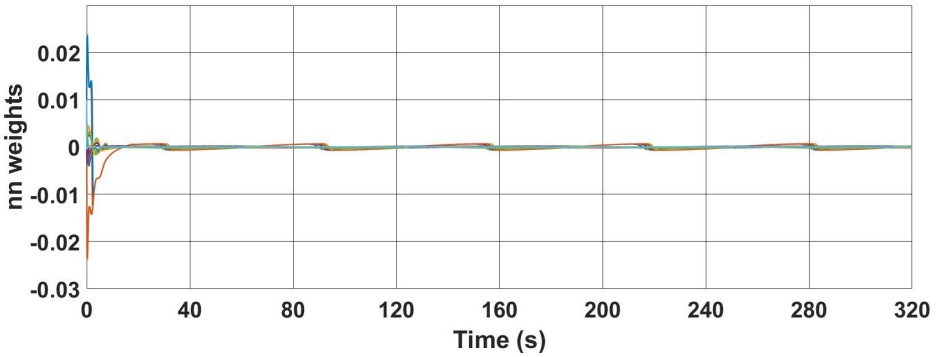

Figure 23. Change in neural-network weights for Scenario 2.

5. Conclusions

A neural-network-based finite-time sliding-mode controller was proposed to control a quadrotor UAV carrying a suspended payload with parametric uncertainties and external disturbances. After deriving the full nonlinear mathematical model of system, the proposed controller was developed, and three different simulations were performed to demonstrate the effectiveness of the proposed controller. First, the system was controlled with a classical SMC. In this simulation, errors in the trajectory tracking in all three axes due to the effects of unknown dynamics and serious chattering effects were observed in the control signals. Very high control signals were also needed to control the system. In the second simulation, a finite-time control structure was added to the SMC to reduce the values of the control signals and the trajectory tracking errors to converge to zero in a finite time. Although FTSMC improved the settling times and the values of the controller signals, it was not effective in eliminating the chattering effect. A neural-network structure was added to FTSMC to construct the proposed controller and effectively handle chattering. With the use of the neural network, chattering disappeared completely, the control signs were significantly reduced, and the settling times were further improved. Future work is to improve the finite-time stability to fixed-time stability to be able to free it of initial conditions to calculate the reaching time, demonstrating the effectiveness of the proposed controller with a real-time experiment.

Author Contributions: The article was written as part of the doctoral thesis of Ö.B. under the supervision of H.M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Ding, Y.; Xin, B.; Chen, J. A Review of Recent Advances in Coordination between Unmanned Aerial and Ground Vehicles. *Unmanned Syst.* **2021**, *9*, 97–117. [CrossRef]
2. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]
3. Cai, G.; Dias, J.; Seneviratne, L. A Survey of Small-Scale Unmanned Aerial Vehicles: Recent Advances and Future Development Trends. *Unmanned Syst.* **2014**, *2*, 175–199. [CrossRef]
4. Shraim, H.; Awada, A.; Youness, R. A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control. *IEEE Aerosp. Electron. Syst. Mag.* **2018**, *33*, 14–33. [CrossRef]
5. Faust, A.; Palunko, I.; Cruz, P.; Fierro, R.; Tapia, L. Automated aerial suspended cargo delivery through reinforcement learning. *Artif. Intell.* **2017**, *247* [CrossRef]

6. Kim, S.; Choi, S.; Kim, H.J. Aerial manipulation using a quadrotor with a two DOF robotic arm. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 4990–4995. [CrossRef]
7. Vahdanipour, M.; Khodabandeh, M. Adaptive fractional order sliding mode control for a quadrotor with a varying load. *Aerosp. Sci. Technol.* **2019**, *86*, 737–747. [CrossRef]
8. Sreenath, K.; Taeyoung Lee.; Kumar, V. Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load. In Proceedings of the IEEE 52nd Annual Conference on Decision and Control (CDC 2013), Firenze, Italy, 10–13 December 2013; Volume 2019, pp. 2269–2274. [CrossRef]
9. Pizetta, I.H.B.; Brandao, A.S.; Sarcinelli-Filho, M. Modelling and control of a PVTOL quadrotor carrying a suspended load. In Proceedings of the ICUAS 2015—International Conference on Unmanned Aircraft Systems, Denver, CO, USA, 9–12 June 2015, [CrossRef]
10. Tang, S.; Kumar, V. Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2216–2222. [CrossRef]
11. Pounds, P.E.; Bersak, D.R.; Dollar, A.M. Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control. *Auton. Robots* **2012**, *33*, 129–142. [CrossRef]
12. Klausen, K.; Fossen, T.I.; Johansen, T.A. Nonlinear control of a multirotor UAV with suspended load. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA , 9–12 June 2015. [CrossRef]
13. Lee, T. Geometric Control of Quadrotor UAVs Transporting a Cable-Suspended Rigid Body. *IEEE Trans. Control. Syst. Technol.* **2018**, *26*, 255–264. [CrossRef]
14. Yang, S.; Xian, B. Energy-Based Nonlinear Adaptive Control Design for the Quadrotor UAV System With a Suspended Payload. *IEEE Trans. Ind. Electron.* **2020**, *67*, 2054–2064. [CrossRef]
15. Qian, L.; Liu, H.H. Path-Following Control of A Quadrotor UAV with A Cable-Suspended Payload Under Wind Disturbances. *IEEE Trans. Ind. Electron.* **2020**, *67*, 2021–2029. [CrossRef]
16. Wang, C.; Song, B.; Huang, P.; Tang, C.; Wang, C.; Song, B.; Huang, P.; Tang, C. Trajectory Tracking Control for Quadrotor Robot Subject to Payload Variation and Wind Gust Disturbance. *J. Intell. Robot Syst.* **2016**, *83*, 315–333. [CrossRef]
17. Bhat, S.P.; Bernstein, D.S. Finite-Time Stability of Continuous Autonomous Systems. *SIAM J. Control Optim.* **2000**, *38*, 751–766. [CrossRef]
18. Sun, K.; Qiu, J.; Karimi, H.R.; Gao, H. A Novel Finite-Time Control for Nonstrict Feedback Saturated Nonlinear Systems With Tracking Error Constraint. *IEEE Trans. Syst. Man, Cybern. Syst.* **2021**, *51*, 3968–3979. [CrossRef]
19. Sui, S.; Chen, C.L.P.; Tong, S. Fuzzy Adaptive Finite-Time Control Design for Nontriangular Stochastic Nonlinear Systems. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 172–184. [CrossRef]
20. Wang, N.; Karimi, H.R.; Li, H.; Su, S.F. Accurate Trajectory Tracking of Disturbed Surface Vehicles: A Finite-Time Control Approach. *IEEE/ASME Trans. Mechatronics* **2019**, *24*, 1064–1074. [CrossRef]
21. Du, P.; Liang, H.; Zhao, S.; Ahn, C.K. Neural-Based Decentralized Adaptive Finite-Time Control for Nonlinear Large-Scale Systems With Time-Varying Output Constraints. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 3136–3147. [CrossRef]
22. Utkin, V.I. Survey Paper: Variable Structure Systems with Sliding Modes. *IEEE Trans. Automat. Contr.* **1977**. [CrossRef]
23. Muliadi, J.; Kusumoputro, B. Neural Network Control System of UAV Altitude Dynamics and Its Comparison with the PID Control System. *J. Adv. Transp.* **2018**, *2018*, 3823201. [CrossRef]
24. Slotine, J.J.; Sastry, S.S. Tracking Control of Nonlinear Systems using Sliding Surfaces. *Int. J. Control* **1983**, *38*, 465–492. [CrossRef]
25. Levant, A. Principles of 2-sliding mode design. *Automatica* **2007**, *43*, 576–586. [CrossRef]
26. Plestan, F.; Glumineau, A.; Laghrouche, S. A new algorithm for high-order sliding mode control. *Int. J. Robust Nonlinear Control.* **2008**, *18*, 441–453. [CrossRef]
27. Frikha, S.; Djemel, M.; Derbel, N. A New Adaptive Neuro-sliding Mode Control for Gantry Crane. *Int. J. Control. Autom. Syst.* **2018**, *16*, 559–565. [CrossRef]
28. Guzey, H.M.; Dierks, T.; Jagannathan, S.; Acar, L. Modified Consensus-based Output Feedback Control of Quadrotor UAV Formations Using Neural Networks. *J. Intell. Robot. Syst. Theory Appl.* **2019**. [CrossRef]
29. Razmi, H.; Afshinfar, S. Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV. *Aerosp. Sci. Technol.* **2019**, *91*, 12–27. [CrossRef]
30. Bansal, S.; Akametalu, A.K.; Jiang, F.J.; Laine, F.; Tomlin, C.J. Learning quadrotor dynamics using neural network for flight control. In Proceedings of the IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016, [CrossRef]
31. Li, S.; Wang, Y.; Tan, J.; Zheng, Y. Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft. *Neurocomputing* **2016**, *216*, 126–134. [CrossRef]
32. Lin, X.; Wang, Y.; Liu, Y. Neural-network-based robust terminal sliding-mode control of quadrotor. *Asian J. Control* **2022**, *24*, 427–438. [CrossRef]
33. Wang, L.; Chai, T.; Zhai, L. Neural-network-based terminal sliding-mode control of robotic manipulators including actuator dynamics. *IEEE Trans. Ind. Electron.* **2009**, *56*, 3296–3304. [CrossRef]

34. Bingöl, Ö.; Güzey, H.M. Neuro sliding mode control of quadrotor UAVs carrying suspended payload. *Adv. Robot.* **2021**, *35*, 255–266. [CrossRef]
35. Cabecinhas, D.; Cunha, R.; Silvestre, C. A trajectory tracking control law for a quadrotor with slung load. *Automatica* **2019**, *106*, 384–389. [CrossRef]
36. Yu, G.; Cabecinhas, D.; Cunha, R.; Silvestre, C. Nonlinear Backstepping Control of a Quadrotor-Slung Load System. *IEEE/ASME Trans. Mechatronics* **2019**, *24*, 2304–2315. [CrossRef]
37. Mofid, O.; Mobayen, S.; Wong, W.K. Adaptive terminal sliding mode control for attitude and position tracking control of quadrotor UAVs in the existence of external disturbance. *IEEE Access* **2020**, *9*, 3428–3440. [CrossRef]
38. Lv, Z.; Zhao, Q.; Li, S.; Wu, Y. Finite-time control design for a quadrotor transporting a slung load. *Control. Eng. Pract.* **2022**, *122*, 105082. [CrossRef]
39. Kang, B.; Miao, Y.; Liu, F.; Duan, J.; Wang, K.; Jiang, S. A second-order sliding mode controller of quad-rotor UAV based on PID sliding mode surface with unbalanced load. *J. Syst. Sci. Complex.* **2021**, *34*, 520–536. [CrossRef]
40. Zhu, Z.; Xia, Y.; Fu, M. Attitude stabilization of rigid spacecraft with finite-time convergence. *Int. J. Robust Nonlinear Control* **2011**, *21*, 686–702. [CrossRef]
41. Lv, W.; Wang, F. Finite-time adaptive fuzzy tracking control for a class of nonlinear systems with unknown hysteresis. *Int. J. Fuzzy Syst.* **2018**, *20*, 782–790. [CrossRef]
42. Hardy, G.H.; Littlewood, J.E.; Pólya, G.; Pólya, G. *Inequalities*; Cambridge University Press: Cambridge, UK, 1952.
43. Xu, R.; Özgüner, Ü. Sliding mode control of a class of underactuated systems. *Automatica* **2008**, *44*, 233–241. [CrossRef]
44. Liu, J.; Gai, W.; Zhang, J.; Li, Y. Nonlinear Adaptive Backstepping with ESO for the Quadrotor Trajectory Tracking Control in the Multiple Disturbances. *Int. J. Control. Autom. Syst.* **2019**, *17*, 2754–2768. [CrossRef]
45. Xiong, J.J.; Zhang, G. Sliding mode control for a quadrotor UAV with parameter uncertainties. In Proceedings of the 2nd International Conference on Control, Automation and Robotics (ICCAR), Hong Kong, 28–30 April 2016. [CrossRef]
46. Feng, Y.; Rabbath, C.A.; Su, C.Y. Modeling of a Micro UAV with Slung Payload. In *Handbook of Unmanned Aerial Vehicles*; Springer: Berlin/Heidelberg, Germany, 2015. [CrossRef]
47. Lewis, F.L. Neural network control of robot manipulators. *IEEE Expert. Syst. their Appl.* **1996**, *11*, 64–75. [CrossRef]
48. Dierks, T.; Jagannathan, S. Output feedback control of a quadrotor UAV using neural networks. *IEEE Trans. Neural Netw.* **2010**, *21*, 50–56. [CrossRef]
49. Riani, A.; Madani, T.; Benallegue, A.; Djouani, K. Adaptive integral terminal sliding mode control for upper-limb rehabilitation exoskeleton. *Control Eng. Pract.* **2018**, *75*, 108–117. [CrossRef]
50. Lian, S.; Meng, W.; Shao, K.; Zheng, J.; Zhu, S.; Li, H. Full Attitude Control of a Quadrotor Using Fast Non-singular Terminal Sliding Mode with Angular Velocity Planning. *IEEE Trans. Ind. Electron.* **2022**, *69*, 1597–1607. [CrossRef]
51. Zheng, E.H.; Xiong, J.J.; Luo, J.L. Second order sliding mode control for a quadrotor UAV. *ISA Trans.* **2014**, *53*, 1350–1356. [CrossRef] [PubMed]
52. Fallaha, C.J.; Saad, M.; Kanaan, H.Y.; Al-Haddad, K. Sliding-mode robot control with exponential reaching law. *IEEE Trans. Ind. Electron.* **2011**, *58*, 600–610. [CrossRef]
53. Polyakov, A. Nonlinear Feedback Design for Fixed-Time Stabilization of Linear Control Systems. *IEEE Trans. Autom. Control.* **2012**, *57*, 2106–2110. [CrossRef]