actuators

MDPI

Article

Intelligent Robust Motion Control of Aerial Robot

Cao-Tri Dinh [1,2], Thien-Dinh Nguyen [1,2], Young-Bok Kim [1], Thinh Huynh [3,*] and Jung-Suk Park [4,*]

[1] Department of Intelligent Robot Engineering, Pukyong National University, Busan 48513, Republic of Korea; tridc@hcmute.edu.vn (C.-T.D.); dinhnt@hcmute.edu.vn (T.-D.N.); kpjiwoo@pknu.ac.kr (Y.-B.K.)
[2] Faculty of Vehicle and Energy Engineering, Ho Chi Minh City University of Technology and Education, Ho Chi Minh City 700000, Vietnam
[3] Department of Smart Mobility Engineering, College of Information Technology and Convergence, Pukyong National University, Busan 48513, Republic of Korea
[4] Training Ship Baek-Kyung, Pukyong National University, Busan 48513, Republic of Korea
* Correspondence: thinhhuynh@pknu.ac.kr (T.H.); pjswith87@pknu.ac.kr (J.-S.P.)

Abstract: This study presents the design of an intelligent robust controller for the 3-degree-of-freedom motion of an aerial robot using waterpower. The proposed controller consists of two parts: (1) an anti-windup super-twisting algorithm that provides stability to the system under actuator saturation; and (2) a fully adaptive radial basis function neural network that estimates and compensates for unexpected influences, i.e., system uncertainties, water hose vibration, and external disturbances. The stability of the entire closed-loop system is analyzed using the Lyapunov stability theory. The controller parameters are optimized such that the effect of these unexpected influences on the control system is minimized. This optimization problem is interpreted in the form of an eigenvalue problem, which is solved using the method of centers. Experiments are conducted where a proportional-integral-derivative controller and a conventional sliding mode controller are deployed for comparison. The results demonstrate that the proposed control system outperforms the others, with small tracking errors and strong robustness against unexpected influences.

Keywords: aerial robot; anti-windup scheme; intelligent control; radial basis function neural network; super-twisting algorithm

check for updates

Academic Editor: Guanghong Yang

Received: 27 March 2025
Revised: 16 April 2025
Accepted: 17 April 2025
Published: 18 April 2025

Citation: Dinh, C.-T.; Nguyen, T.-D.; Kim, Y.-B.; Huynh, T.; Park, J.-S. Intelligent Robust Motion Control of Aerial Robot. *Actuators* **2025**, *14*, 197. https://doi.org/10.3390/act14040197

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Unmanned water-powered aerial robotic systems have potential for specialized missions related to bodies of water, e.g., observation, exploration, rescue, and firefighting [1]. As shown in Figure 1, these systems use the pressurized water delivered from a water pump through a flexible hose to generate the water thrust, propelling the robot. In most missions mentioned, the robot first needs to fly to the desired position before performing the tasks. As the first step to address this motion control problem, numerous simple controllers, e.g., derivative (D) [2–5], proportional-derivative (PD) [1,6], and proportional-integral-derivative (PID) [7] controllers, were used in practice. Unfortunately, the control performance provided by these controllers generally failed to fulfill the expectations. The main factors contributing to this result come from influences such as physical limitations of the actuators, system uncertainties, external disturbances, and water hose vibration.

The considered aerial robot powered by water shares certain characteristics with a drone-slung load system. As part of studies addressing motion control of such systems, Lee et al. [8] designed a new trajectory generator and an effective antisway tracking controller for a drone with a cable-suspended payload. Akhtar et al. [9] designed

a path-invariant controller for a quadrotor suspended by a taut cable using a global parameterization of the underlying manifold. Goodman et al. [10] proposed a geometric trajectory tracking controller for the cooperative task of four quadrotor UAVs transporting a rigid body load via inflexible elastic cables. Moreover, several related studies [11,12] have also investigated the drone transportation with a cable-suspended payload. These studies focused on eliminating the residual swing of the suspended load and/or driving the load to follow the predefined trajectories. In contrast, tracking motion control of the robot itself, not the flexible hose, is the main target in our study, especially in consideration of the robot's specialized applications mentioned previously.



(**a**) (**b**)

Figure 1. Flight motions of unmanned aerial robotic systems in real-world tests: (**a**) Unmanned water-powered aerial vehicles [1]; (**b**) Dragon Firefighter [5].

Among studies related to water-powered aerial robots, Maezawa et al. [13] designed an H_∞ controller to suppress the robot's oscillations caused by the mobile base. Nevertheless, the vibration is still significant, and the limitations of the actuators heavily affect the system's stability. Tri et al. [14] proposed an extended state observer-based super-twisting sliding mode control to regulate the robot's angular motion. However, actuator saturation and system uncertainties were not considered in this study. Other robust nonlinear controllers, e.g., sliding mode controller (SMC) [15], and robust composite controller [16] have also been proposed. Their effectiveness, unfortunately, has not been validated in practical testing. In addition, the control parameters of these studies were not optimized, i.e., they were chosen by trial-and-error methods. Overall, these studies fail to comprehensively address the unexpected influences mentioned above.

In recent years, several effective strategies have been adopted. For instance, Zhuang Liu et al. [17] have proposed an adaptive disturbance observer-based fixed-time backstepping control algorithm for uncertain robotic systems. However, an adaptive sliding mode disturbance observer in this study relies on exact system dynamics, i.e., a perfect knowledge of the nonlinear elements of the robotic system. Therefore, if the actual system behaves differently than expected in reality, for example, due to modelling inaccuracies and/or poor measurement quality, the control performance can be significantly degraded. Furthermore, several studies [18–22] have also proposed a neural network-based disturbance compensator for robotic systems. However, only the weight matrix was updated, while the center of the expectation values and standard deviations of the Gaussian function are manually chosen.

Motivated to tackle these issues, this research proposes an intelligent robust motion controller for the 3-degree-of-freedom (3-DoF) aerial robot using waterpower. Herein, a first-order anti-windup scheme is deployed to handle the actuator saturation. A super-twisting algorithm is then constructed to ensure system stability. Next, a fully adaptive

radial basis function (RBF) neural network is designed to estimate and compensate for the mentioned influences. As a result, the closed-loop system is stable in the sense of Lyapunov stability theory, and the control errors are bounded even in the presence of unexpected influences. Moreover, the control parameters are optimized such that this boundedness is minimized.

In summary, the main contributions of this study are outlined as follows:

1. For the first time, an anti-windup super-twisting sliding mode controller is designed and implemented for the water-powered aerial robot to ensure its stability, even in the presence of actuator saturation.
2. An adaptive RBF neural network compensates for unexpected influences, in which not only its weights but also the expectation values and standard deviations of the Gaussian function are effectively updated.
3. The stability of the entire system is analyzed by the Lyapunov stability theory.
4. The problem of control parameters optimization is interpreted in the form of an eigenvalue problem (EVP), which is solved by the method of centers.
5. The effectiveness of the proposed system is experimentally validated and evaluated.
6. The rest of this paper is outlined as follows. Firstly, Section 2 introduces the mathematical model of the proposed system to realize its dynamical characteristics. Secondly, Section 3 presents the control system design. The experimental study is then illustrated in Section 4. Finally, Section 5 summarizes the main findings.

2. Mathematical Model

As shown in Figure 2, the aerial robot powered by water includes three main parts, namely a water pump, a flexible water hose, and a head assembly. Water is pressurized by the water pump, then flows to the head assembly through the water hose and is jetted out at the nozzles' outlets, generating thrust. In addition, four nozzles are connected to a water manifold via swivel joints. Each nozzle is driven by a servo motor. Therefore, the robot can achieve 6-DoF motion in the air by regulating the direction of the water thrust through the independently controlled nozzles.

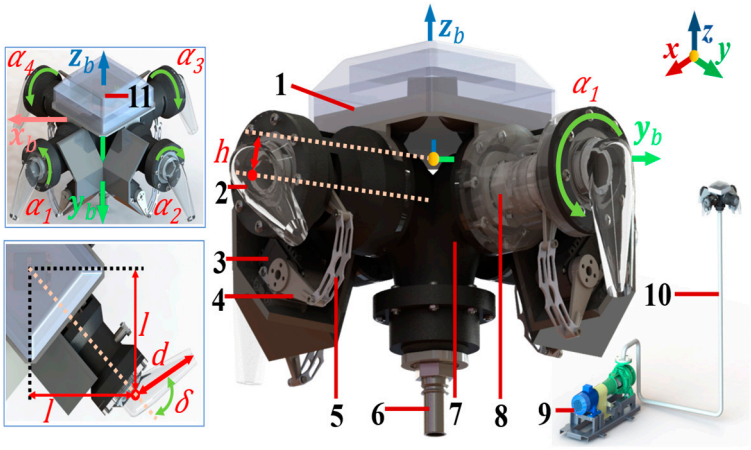

Figure 2. Dimensions and notations of the controlled aerial robot: 1. Circuit box; 2. Nozzle module; 3. Servo motor; 4. Servo arm; 5. Servo linkage; 6. Water inlet port; 7. Water manifold; 8. Swivel joint; 9. Water pump; 10. Water hose; 11. Sensor.

In terms of flight movement, altitude, roll, and pitch represent the fundamental motions of the aerial robot. Together, these motions provide the robot with full controllability over its position in the air. Therefore, as the first step in investigating its basic flight behav-

ior, a mathematical model of the 3-DoF aerial robot has been derived from the author's previous study [7], as follows:

$$\begin{gathered}\ddot{\boldsymbol{x}} = \boldsymbol{L}\boldsymbol{u} + \boldsymbol{g} + \boldsymbol{d}, \\ \boldsymbol{x} = \begin{bmatrix} z & \phi & \theta \end{bmatrix}^T, \ \boldsymbol{g} = \begin{bmatrix} -g & 0 & 0 \end{bmatrix}^T, \ \boldsymbol{d} = [d_z \ d_\phi \ d_\theta]^T, \\ \boldsymbol{u} = [u_z \ u_\phi \ u_\theta]^T = \boldsymbol{Q}[\sin\alpha_1 \ \sin\alpha_2 \ \sin\alpha_3 \ \sin\alpha_4]^T, \\ \boldsymbol{L} = \begin{bmatrix} \cos\phi\cos\theta & 0 & 0 \\ 0 & 1 & \sin\phi\tan\theta \\ 0 & 0 & \cos\phi \end{bmatrix}, \ \boldsymbol{Q} = \begin{bmatrix} q_1 & q_1 & q_1 & q_1 \\ q_2 & q_2 & -q_2 & -q_2 \\ -q_3 & q_3 & q_3 & -q_3 \end{bmatrix}, \\ q_1 = \frac{q_4}{m}, \quad q_2 = \frac{q_4}{J_{bx}} + q_5, \quad q_3 = \frac{q_4}{J_{by}} + q_5, \\ q_4 = \frac{\dot{m}_0^2 \sin\delta}{16\rho a}, \quad q_5 = \frac{\sqrt{2}d\dot{m}_0^2 \sin\delta\cos\delta}{32\rho a} \end{gathered} \tag{1}$$

In Equation (1), z is the altitude of the head unit. ϕ and θ are the roll and pitch Euler angles, respectively, describing the orientation of the head unit with respect to the reference frame. Additionally, α_i (i = 1~4) is the rotation angle of the ith nozzle about its rotational axis. Specifically, the nozzles can be rotated from the horizontal plane corresponding to $\alpha_i = 0$ to the totally downward direction corresponding to $\alpha_i = \frac{\pi}{2}$. In addition, the gravitational acceleration and the mass of the head with its contained water are denoted by g and m. J_{bx} and J_{by} are the moments of inertia with respect to x_b- and y_b-axes. Furthermore, ρ, A, a, δ, and $\dot{m}_0$ denote the water density, the cross-sectional areas of the inlet and outlet ports, the folded angle between the nozzle and the horizontal plane, and the total water mass flow rate, respectively. Finally, $\boldsymbol{d}$ represents the unexpected influences, including the water hose effect, system uncertainties, and external factors.

Remark 1. *In fact, the large-scale water pump causes several issues, such as slow responsiveness. They may lead to a significant delay in water thrust control. Therefore, the proposed solution to overcome this is to keep the water flow rate $\dot{m}_0$ constant. Consequently, the water force distribution of the flying robot depends entirely on the rotation angle of the nozzles.*

3. Control System Design

The objective of the control system is to control the altitude of the robot to follow the desired reference and to stabilize the roll and pitch angles at the origin, even in the presence of influences.

Let $\overline{\boldsymbol{u}} = \begin{bmatrix} \overline{u}_z & \overline{u}_\phi & \overline{u}_\theta \end{bmatrix}^T$ be the output computed by the controller, i.e., the control signal without saturation. As illustrated in Figure 3, the designed law $\overline{\boldsymbol{u}} \in R^{3\times1}$ consists of two parts: an anti-windup super-twisting law $\overline{\boldsymbol{u}}_1 \in R^{3\times1}$ for providing stability to the robot under actuator saturation and an estimation $\overline{\boldsymbol{u}}_2 \in R^{3\times1}$ for the influences' compensation:

$$\overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}_1 + \overline{\boldsymbol{u}}_2, \ \overline{\boldsymbol{u}} = [\overline{u}_z \ \overline{u}_\phi \ \overline{u}_\theta]^T \tag{2}$$

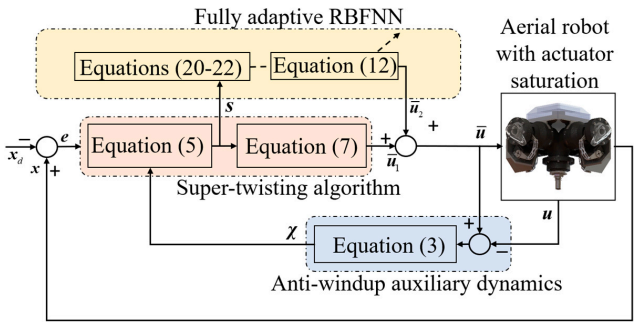

Figure 3. Block diagram of the control system.

3.1. Anti-Windup Super-Twisting Algorithm

The first-order anti-windup dynamics are proposed as:

$$\begin{aligned} \dot{\boldsymbol{\chi}} &= -\boldsymbol{k}\boldsymbol{\chi} + \Delta\boldsymbol{u}, \\ \Delta\boldsymbol{u} &= [\Delta u_z \;\; \Delta u_\phi \;\; \Delta u_\theta]^T = \boldsymbol{L}(\overline{\boldsymbol{u}} - \boldsymbol{u}) \end{aligned} \tag{3}$$

where $\boldsymbol{\chi} = [\chi_z \;\; \chi_\phi \;\; \chi_\theta]^T$, $\boldsymbol{k} \in \mathrm{R}^{3\times3}$. In which, $\boldsymbol{k} > 0$ is a diagonal symmetric matrix.

It is significant to point out that $\boldsymbol{k}$ is chosen from the desired bandwidth of the control system to achieve good compensation performance. Additionally, the relationship between $\boldsymbol{u}$ and $\overline{\boldsymbol{u}}$ can be described as the following:

$$u_\kappa = \begin{cases} \overline{u}_{\kappa_{\min}} & , \overline{u}_\kappa < \overline{u}_{\kappa_{\min}} \\ \overline{u}_\kappa & , \overline{u}_{\kappa_{\min}} \le \overline{u}_\kappa \le \overline{u}_{\kappa_{\max}} \\ \overline{u}_{\kappa_{\max}} & , \overline{u}_\kappa > \overline{u}_{\kappa_{\max}} \end{cases}, \quad \kappa = z, \phi, \theta \tag{4}$$

where $\overline{u}_{\kappa_{\min}}$ and $\overline{u}_{\kappa_{\max}}$ represent the lower and upper bounds of forces and torques acting on the head assembly. Their values are calculated from the limited operating range of the rotating nozzles. Moreover, once a stable control $\boldsymbol{u}$ is derived, Δu_κ is guaranteed to be bounded, assuming by a constant $\Delta u_{\kappa_{\max}} > 0$.

Initially, let us define the error between the desired reference $\boldsymbol{x}_d = [z_d \quad 0 \quad 0]^T$ and the plant output $\boldsymbol{x}$ as $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$. Assuming z_d be twice continuously differentiable, a sliding manifold is designed as follows:

$$\boldsymbol{s} = [s_z \;\; s_\phi \;\; s_\theta]^T = \dot{\boldsymbol{e}} + \boldsymbol{n}_1\boldsymbol{e} + \boldsymbol{n}_2\int_0^t \boldsymbol{e}(\iota)d\iota - \boldsymbol{\chi} \tag{5}$$

where $\boldsymbol{n}_1, \boldsymbol{n}_2 \in R^{3\times3}$, $\boldsymbol{n}_1 > 0$, and $\boldsymbol{n}_2 > 0$ are diagonal gain matrices.

The time derivative of the sliding manifold can be derived as:

$$\dot{\boldsymbol{s}} = \ddot{\boldsymbol{x}}_d - \boldsymbol{L}\overline{\boldsymbol{u}} - \boldsymbol{g} - \boldsymbol{d} + \boldsymbol{n}_1\dot{\boldsymbol{e}} + \boldsymbol{n}_2\boldsymbol{e} + \boldsymbol{k}\boldsymbol{\chi} \tag{6}$$

The structure of the anti-windup super-twisting controller (AWSTC) is proposed by:

$$\begin{aligned} \overline{\boldsymbol{u}}_1 &= \boldsymbol{L}^{-1}\left(\ddot{\boldsymbol{x}}_d - \boldsymbol{g} + \boldsymbol{n}_1\dot{\boldsymbol{e}} + \boldsymbol{n}_2\boldsymbol{e} + \boldsymbol{k}\boldsymbol{\chi}\right) \\ &+\boldsymbol{L}^{-1}\left[n_{3z}|s_z|^{\frac{1}{2}}\mathrm{sgn}(s_z) \;\; n_{3\phi}|s_\phi|^{\frac{1}{2}}\mathrm{sgn}(s_\phi) \;\; n_{3\theta}|s_\theta|^{\frac{1}{2}}\mathrm{sgn}(s_\theta)\right]^T \\ &+\boldsymbol{L}^{-1}\left[n_{4z}\int_0^t \mathrm{sgn}(s_z(\iota))d\iota \;\; n_{4\phi}\int_0^t \mathrm{sgn}(s_\phi(\iota))d\iota \;\; n_{4\theta}\int_0^t \mathrm{sgn}(s_\theta(\iota))d\iota\right]^T \end{aligned} \tag{7}$$

with $n_{3\kappa}$ and $n_{4\kappa}$ are the positive gains ($\kappa = z, \; \phi, \; \theta$).

3.2. RBF Neural Network

According to the universal approximation theorem [23], each disturbance $d_\kappa \in R$ can be expressed as follows:

$$d_\kappa = \boldsymbol{w}_\kappa\boldsymbol{h}_\kappa + \varepsilon_\kappa, \quad \kappa = z, \; \phi, \; \theta \tag{8}$$

where:

$$\boldsymbol{w}_\kappa = \begin{bmatrix} w_{\kappa1} & \cdots & w_{\kappa j} \end{bmatrix} \tag{9}$$

$$\begin{aligned} \boldsymbol{h}_\kappa &= \begin{bmatrix} h_{\kappa1} & \cdots & h_{\kappa j} \end{bmatrix}^T \\ &= \left[\exp\left(-(s_\kappa - \sigma_{\kappa1})^2/\beta_\kappa^2\right) \;\; \cdots \;\; \exp\left(-(s_\kappa - \sigma_{\kappa j})^2/\beta_\kappa^2\right)\right]^T \end{aligned} \tag{10}$$

In the above equations, j is the number of neurons, $\boldsymbol{w}_\kappa \in R^{1\times j}$ is the ideal weight vector, $\boldsymbol{h}_\kappa \in R^{j\times 1}$ is the vector of the Gaussian activation functions. Moreover, the scalars $\sigma_{\kappa j}$ and $\beta_{\kappa j}$ are the ideal expectation value and standard deviation of each function. ε_κ is a small approximation error of the neural network, and it is bounded. Additionally, let the estimation of disturbance $\hat{d}_\kappa \in R$ be proposed as:

$$\hat{d}_\kappa = \hat{\boldsymbol{w}}_\kappa \hat{\boldsymbol{h}}_\kappa \tag{11}$$

where $\hat{\boldsymbol{w}}_\kappa = \begin{bmatrix} \hat{w}_{\kappa 1} & \cdots & \hat{w}_{\kappa j} \end{bmatrix}$ and $\hat{\boldsymbol{h}}_\kappa = \begin{bmatrix} \hat{h}_{\kappa 1} & \cdots & \hat{h}_{\kappa j} \end{bmatrix}^T$ represent the estimation of $\boldsymbol{w}_\kappa$ and $\boldsymbol{h}_\kappa$, respectively.

Besides, $\hat{\sigma}_{\kappa j}$ and $\hat{\beta}_{\kappa j}$ denote the estimated value of $\sigma_{\kappa j}$ and $\beta_{\kappa j}$ for the corresponding $\hat{h}_{\kappa j}$. Thereby, $\overline{\boldsymbol{u}}_2$ is designed as follows:

$$\begin{aligned} \overline{\boldsymbol{u}}_2 &= -\boldsymbol{L}^{-1}\hat{\boldsymbol{d}}, \\ \hat{\boldsymbol{d}} = \begin{bmatrix} \hat{d}_z & \hat{d}_\phi & \hat{d}_\theta \end{bmatrix}^T &= \begin{bmatrix} \hat{\boldsymbol{w}}_z \hat{\boldsymbol{h}}_z & \hat{\boldsymbol{w}}_\phi \hat{\boldsymbol{h}}_\phi & \hat{\boldsymbol{w}}_\theta \hat{\boldsymbol{h}}_\theta \end{bmatrix}^T \end{aligned} \tag{12}$$

It should be highlighted that if $\hat{d}_\kappa$ can properly follow d_κ, $\overline{\boldsymbol{u}}_2$ can compensate for the influences effectively.

Finally, by substituting (7) and (12) into (2), the corresponding nozzle control, $\overline{\boldsymbol{\alpha}} = \begin{bmatrix} \overline{\alpha}_1 & \overline{\alpha}_2 & \overline{\alpha}_3 & \overline{\alpha}_4 \end{bmatrix}^T$, is computed from the following:

$$\sin \overline{\boldsymbol{\alpha}} = \boldsymbol{Q}^\dagger \overline{\boldsymbol{u}} \tag{13}$$

with $\boldsymbol{Q}^\dagger \in R^{4\times 3}$ is the right inverse of $\boldsymbol{Q}$ and is derived by $\boldsymbol{Q}^\dagger = \boldsymbol{Q}^T\left(\boldsymbol{Q}\boldsymbol{Q}^T\right)^{-1}$.

3.3. Update Laws for Parameters of RBF Neural Network

Let us define the difference between the ideal and the estimated parameters of the weight and Gaussian function as $\widetilde{\boldsymbol{h}}_\kappa = \boldsymbol{h}_\kappa - \hat{\boldsymbol{h}}_\kappa$ and $\widetilde{\boldsymbol{w}}_\kappa = \boldsymbol{w}_\kappa - \hat{\boldsymbol{w}}_\kappa$, respectively. Consequently, the approximation error of the disturbance can be expressed as follows:

$$\widetilde{d}_\kappa = d_\kappa - \hat{d}_\kappa = \hat{\boldsymbol{w}}_\kappa \widetilde{\boldsymbol{h}}_\kappa + \widetilde{\boldsymbol{w}}_\kappa \hat{\boldsymbol{h}}_\kappa + \widetilde{\boldsymbol{w}}_\kappa \widetilde{\boldsymbol{h}}_\kappa + \varepsilon_\kappa \tag{14}$$

To provide a full parameter update law, let us expand $h_\kappa \in R$ following the first-order Taylor approximation at $\sigma_{\kappa j} = \hat{\sigma}_{\kappa j}$ and $\beta_\kappa = \hat{\beta}_\kappa$:

$$\begin{aligned} h_{\kappa j} &= h_{\kappa j}\Big|_{\substack{\sigma_{\kappa j}=\hat{\sigma}_{\kappa j} \\ \beta_\kappa = \hat{\beta}_\kappa}} + \frac{\partial h_{\kappa j}}{\partial \sigma_{\kappa j}}\Big|_{\substack{\sigma_{\kappa j}=\hat{\sigma}_{\kappa j} \\ \beta_\kappa = \hat{\beta}_\kappa}} \widetilde{\sigma}_{\kappa j} + \frac{\partial h_{\kappa j}}{\partial \beta_\kappa}\Big|_{\substack{\sigma_{\kappa j}=\hat{\sigma}_{\kappa j} \\ \beta_\kappa = \hat{\beta}_\kappa}} \widetilde{\beta}_\kappa + \Delta_{\kappa j} \\ &= \hat{h}_{\kappa j} + \frac{2\left(s_\kappa - \hat{\sigma}_{\kappa j}\right)}{\hat{\beta}_\kappa^2}\hat{h}_{\kappa j}\widetilde{\sigma}_{\kappa j} - \frac{2\left(s_\kappa - \hat{\sigma}_{\kappa j}\right)^2}{\hat{\beta}_\kappa^3}\hat{h}_{\kappa j}\widetilde{\beta}_\kappa + \Delta_{\kappa j} \end{aligned} \tag{15}$$

where $\widetilde{\sigma}_{\kappa j} = \sigma_{\kappa j} - \hat{\sigma}_{\kappa j}$, $\widetilde{\beta}_\kappa = \beta_\kappa - \hat{\beta}_\kappa$, and $\Delta_{\kappa j}$ represents the remaining high-order terms.

By substituting (15) into (14), the approximation error of the disturbance can be interpreted as:

$$\widetilde{d}_\kappa = \widetilde{\boldsymbol{\sigma}}_\kappa \boldsymbol{\Gamma}_\kappa \hat{\boldsymbol{w}}_\kappa^T + \widetilde{\beta}_\kappa \hat{\boldsymbol{w}}_\kappa \boldsymbol{\Lambda}_\kappa + \widetilde{\boldsymbol{w}}_\kappa \hat{\boldsymbol{h}}_\kappa + \bar{\varepsilon}_\kappa \tag{16}$$

where:

$$\begin{gathered}\widetilde{\boldsymbol{\sigma}}_\kappa = \left[\widetilde{\sigma}_{\kappa 1} \ \cdots \ \widetilde{\sigma}_{\kappa j}\right], \boldsymbol{\Delta}_\kappa = \left[\Delta_{\kappa 1} \ \cdots \ \Delta_{\kappa j}\right]^T, \\ \bar{\varepsilon}_\kappa = \hat{\boldsymbol{w}}_\kappa \boldsymbol{\Delta}_\kappa + \widetilde{\boldsymbol{w}}_\kappa \widetilde{\boldsymbol{h}}_\kappa + \varepsilon_\kappa, \\ \boldsymbol{\Gamma}_\kappa = diag\left\{2(s_\kappa - \hat{\sigma}_{\kappa 1})\hat{\beta}_\kappa^{-2}\hat{h}_{\kappa 1}, \ \cdots, 2(s_\kappa - \hat{\sigma}_{\kappa j})\hat{\beta}_\kappa^{-2}\hat{h}_{\kappa j}\right\}, \\ \boldsymbol{\Lambda}_\kappa = \left[-2(s_\kappa - \hat{\sigma}_{\kappa 1})^2\hat{\beta}_\kappa^{-3}\hat{h}_{\kappa 1} \quad \cdots \quad -2(s_\kappa - \hat{\sigma}_{\kappa j})^2\hat{\beta}_\kappa^{-3}\hat{h}_{\kappa j}\right]^T\end{gathered} \tag{17}$$

with $\boldsymbol{\Gamma}_\kappa \in R^{j\times j}, \boldsymbol{\Lambda}_\kappa \in R^{j\times 1}, \widetilde{\boldsymbol{\sigma}}_\kappa \in R^{1\times j}, \widetilde{\beta}_\kappa \in \mathrm{R}, \boldsymbol{\Delta}_\kappa \in R^{j\times 1}$.

It is notable that the term $\bar{\varepsilon}_\kappa$ is considered as the total approximation error and it is bounded [24], assuming by $\bar{\varepsilon}_{\kappa_{\max}} > 0$. By substituting (7), (12) into (2) and then (2) into (6), the dynamics of the sliding manifold can be obtained as:

$$\begin{cases} \dot{s}_\kappa = -n_{3\kappa}|s_\kappa|^{\frac{1}{2}}\mathrm{sgn}(s_\kappa) + v_\kappa - \widetilde{d}_\kappa \\ \dot{v}_\kappa = -n_{4\kappa}\mathrm{sgn}(s_\kappa) \end{cases} \tag{18}$$

The right-hand side of (18) is discontinuous, its solutions can be understood in the sense of Filippov [25]. Additionally, let us define $z_\kappa = \begin{bmatrix} z_{1\kappa} & z_{2\kappa} \end{bmatrix}^T = \begin{bmatrix} |s_\kappa|^{\frac{1}{2}}\mathrm{sgn}(s_\kappa) & v_\kappa \end{bmatrix}^T$. Then, its time derivative can be expressed as the following:

$$\begin{gathered}\dot{\boldsymbol{z}}_\kappa = |s_\kappa|^{\frac{-1}{2}}\boldsymbol{A}_\kappa \boldsymbol{z}_\kappa + |s_\kappa|^{\frac{-1}{2}}\widetilde{d}_\kappa \boldsymbol{B}_\kappa, \\ \boldsymbol{A}_\kappa = \frac{1}{2}\begin{bmatrix} -n_{3\kappa} & 1 \\ -2n_{4\kappa} & 0 \end{bmatrix}, \ \boldsymbol{B}_\kappa = \frac{1}{2}\begin{bmatrix} -1 \\ 0 \end{bmatrix}\end{gathered} \tag{19}$$

Moreover, by considering $\boldsymbol{P}_\kappa = \boldsymbol{P}_\kappa^T > 0 \in R^{2\times 2}$, a fully updated law of the RBF neural network can be designed by:

$$\dot{\hat{\boldsymbol{w}}}_\kappa = \begin{cases} \eta_\kappa |s_\kappa|^{\frac{-1}{2}}\left(\hat{\boldsymbol{h}}_\kappa \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa\right)^T & if \ \ |s_\kappa| > \lambda_\kappa \\ \boldsymbol{O}_{1\times j} & otherwise \end{cases} \tag{20}$$

$$\dot{\hat{\boldsymbol{\sigma}}}_\kappa = \begin{cases} \mu_\kappa |s_\kappa|^{\frac{-1}{2}}\left(\boldsymbol{\Gamma}_\kappa \hat{\boldsymbol{w}}_\kappa^T \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa\right)^T & if \ \ |s_\kappa| > \lambda_\kappa \\ \boldsymbol{O}_{1\times j} & otherwise \end{cases} \tag{21}$$

$$\dot{\hat{\beta}}_\kappa = \begin{cases} v_\kappa |s_\kappa|^{\frac{-1}{2}}\left(\hat{\boldsymbol{w}}_\kappa \boldsymbol{\Lambda}_\kappa \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa\right)^T & if \ \ |s_\kappa| > \lambda_\kappa \\ 0 & otherwise \end{cases} \tag{22}$$

where $\boldsymbol{O}_{m\times n}$ represents the m-by-n matrix of zero. η_κ, μ_κ, and $v_\kappa > 0$ are the learning rates of the corresponding $\hat{\boldsymbol{w}}_\kappa$, $\hat{\sigma}_\kappa$, and $\hat{\beta}_\kappa$, respectively. λ_κ is the activation threshold, which will be determined later. It is significant to point out that λ_κ must be greater than zero to avoid the singularity of the update law due to the term $|s_\kappa|^{\frac{-1}{2}}$.

3.4. Stability Analysis

Let us consider a Lyapunov function candidate as the following:

$$V_\kappa = V_{\kappa_1} + V_{\kappa_2} + V_{\kappa_3} + V_{\kappa_4} + V_{\kappa_5} \tag{23}$$

where:

$$V_{\kappa_1} = \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{z}_\kappa, \ V_{\kappa_2} = \frac{1}{\eta_\kappa}\widetilde{\boldsymbol{w}}_\kappa \widetilde{\boldsymbol{w}}_\kappa^T, \ V_{\kappa_3} = \frac{1}{\mu_\kappa}\widetilde{\boldsymbol{\sigma}}_\kappa \widetilde{\boldsymbol{\sigma}}_\kappa^T, \ V_{\kappa_4} = \frac{1}{v_\kappa}\widetilde{\beta}_\kappa^2, \ V_{\kappa_5} = \frac{1}{2}\chi_\kappa^2$$

with $\boldsymbol{P}_\kappa = \boldsymbol{P}_\kappa^T = \frac{1}{2}\begin{bmatrix} 4n_{4\kappa} & -n_{3\kappa} \\ -n_{3\kappa} & 4 \end{bmatrix} > 0.$

Remark 2. *V_κ is continuously differentiable, except on $s_\kappa = 0$. However, one sees that the trajectories of (19) just cross the sliding surface and cannot stay on it, except when the origin has been reached. This means that V_κ is differentiable almost everywhere till the moment when $[s_\kappa \;\; v_\kappa]^T = 0$ [26].*

Taking the time derivative of (23) and noting that the ideal parameters of the RBF neural network are constant, i.e., $\dot{\sigma}_{\kappa j} = \dot{\beta}_{\kappa j} = \dot{w}_{\kappa j} = 0$, one can have:

$$\begin{aligned}\dot{V}_{\kappa_1} &= \left(\dot{\boldsymbol{z}}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{z}_\kappa + \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \dot{\boldsymbol{z}}_\kappa\right) = |s_\kappa|^{\frac{-1}{2}} \boldsymbol{z}_\kappa^T \left(\boldsymbol{A}_\kappa^T \boldsymbol{P}_\kappa + \boldsymbol{P}_\kappa \boldsymbol{A}_\kappa\right) \boldsymbol{z}_\kappa \\ &+2|s_\kappa|^{\frac{-1}{2}} \left(\widetilde{\sigma}_\kappa \boldsymbol{\Gamma}_\kappa \hat{\boldsymbol{w}}_\kappa^T + \widetilde{\beta}_\kappa \hat{\boldsymbol{w}}_\kappa \boldsymbol{\Lambda}_\kappa + \widetilde{\boldsymbol{w}}_\kappa \hat{\boldsymbol{h}}_\kappa + \bar{\varepsilon}_\kappa\right) \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa\end{aligned} \tag{24}$$

$$\dot{V}_{\kappa_2} = \left(\dot{\widetilde{\boldsymbol{w}}}_\kappa \widetilde{\boldsymbol{w}}_\kappa^T + \widetilde{\boldsymbol{w}}_\kappa \dot{\widetilde{\boldsymbol{w}}}_\kappa^T\right) / \eta_\kappa = -2\widetilde{\boldsymbol{w}}_\kappa \dot{\hat{\boldsymbol{w}}}_\kappa^T / \eta_\kappa \tag{25}$$

$$\dot{V}_{\kappa_3} = \left(\dot{\widetilde{\sigma}}_\kappa \widetilde{\sigma}_\kappa^T + \widetilde{\sigma}_\kappa \dot{\widetilde{\sigma}}_\kappa^T\right) / \mu_\kappa = -2\widetilde{\sigma}_\kappa \dot{\hat{\sigma}}_\kappa^T / \mu_\kappa \tag{26}$$

$$\dot{V}_{\kappa_4} = 2\widetilde{\beta}_\kappa \dot{\widetilde{\beta}}_\kappa / \upsilon_\kappa = -2\widetilde{\beta}_\kappa \dot{\hat{\beta}}_\kappa / \upsilon_\kappa \tag{27}$$

$$\dot{V}_{\kappa_5} = \chi_\kappa \dot{\chi}_\kappa = -\chi_\kappa^2 k_\kappa + \chi_\kappa \Delta u_\kappa \tag{28}$$

Consequently, $\dot{V}_\kappa$ can be obtained as the following:

$$\begin{aligned}\dot{V}_\kappa &= 2\widetilde{\beta}_\kappa \left(|s_\kappa|^{\frac{-1}{2}} \hat{\boldsymbol{w}}_\kappa \boldsymbol{\Lambda}_\kappa \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa - \upsilon_\kappa^{-1} \dot{\hat{\beta}}_\kappa\right) - \chi_\kappa^2 k_\kappa + \chi_\kappa \Delta u_\kappa a \\ &+|s_\kappa|^{\frac{-1}{2}} \boldsymbol{z}_\kappa^T \left(\boldsymbol{A}_\kappa^T \boldsymbol{P}_\kappa + \boldsymbol{P}_\kappa \boldsymbol{A}_\kappa\right) \boldsymbol{z}_\kappa + 2\widetilde{\boldsymbol{w}}_\kappa \left(|s_\kappa|^{\frac{-1}{2}} \hat{\boldsymbol{h}} \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa - \eta_\kappa^{-1} \dot{\hat{\boldsymbol{w}}}_\kappa^T\right) \\ &+2\widetilde{\sigma}_\kappa \left(|s_\kappa|^{\frac{-1}{2}} \boldsymbol{\Gamma}_\kappa \hat{\boldsymbol{w}}_\kappa^T \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa - \mu_\kappa^{-1} \dot{\hat{\sigma}}_\kappa^T\right) + 2|s_\kappa|^{\frac{-1}{2}} \bar{\varepsilon}_\kappa \boldsymbol{z}_\kappa^T \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa\end{aligned} \tag{29}$$

In case of $|s_\kappa| > \lambda_\kappa$, the update law is activated. Substituting (20)–(22) into (29), letting $\vartheta_\kappa \triangleq \left[|s_\kappa|^{\frac{-1}{4}} \boldsymbol{z}_\kappa^T \quad \chi_\kappa\right]^T$ and $\boldsymbol{K}_\kappa = \boldsymbol{A}_\kappa^T \boldsymbol{P}_\kappa + \boldsymbol{P}_\kappa \boldsymbol{A}_\kappa$, one gets:

$$\begin{aligned}\dot{V}_\kappa &= \vartheta_\kappa^T \begin{bmatrix} \boldsymbol{K}_\kappa & \boldsymbol{O}_{2\times 1} \\ \boldsymbol{O}_{1\times 2} & -k_\kappa \end{bmatrix} \vartheta_\kappa + \vartheta_\kappa^T \boldsymbol{Z}_\kappa \begin{bmatrix} \bar{\varepsilon}_\kappa \\ \Delta u_\kappa \end{bmatrix}, \\ \boldsymbol{Z}_\kappa &= \begin{bmatrix} 2\boldsymbol{P}_\kappa \boldsymbol{B}_\kappa & \boldsymbol{O}_{2\times 1} \\ 0 & 1 \end{bmatrix}\end{aligned} \tag{30}$$

By enforcing $n_{3\kappa}^2 = 4n_{4\kappa}$, (30) becomes:

$$\dot{V}_\kappa = -\vartheta_\kappa^T \boldsymbol{F}_\kappa \vartheta_\kappa + \vartheta_\kappa^T \boldsymbol{Z}_\kappa [\bar{\varepsilon}_\kappa \;\; \Delta u_\kappa]^T \tag{31}$$

with $\boldsymbol{F}_\kappa = diag\{n_{3\kappa} n_{4\kappa},\ n_{3\kappa}/2,\ k_\kappa\}$.

One can see $\dot{V}_\kappa \leq 0$ is always held outside the set $E = \{\|\vartheta_\kappa\| < \Omega_\kappa\}$ with $\Omega_\kappa = \max(\bar{\varepsilon}_{\kappa_{\max}}, \Delta u_{\kappa_{\max}}) \|\boldsymbol{Z}_\kappa\| \lambda_{\min}^{-1}(\boldsymbol{F}_\kappa)$. It is worth noting that $\|(\bullet_1)\|$ and $\lambda_{\min}(\bullet_2)$ are the Euclidean norm and the minimum eigenvalue of the corresponding $(\bullet_1)$ and $(\bullet_2)$, respectively. One concludes that the trajectories of ϑ_κ will come into the superset of E in finite time and stay thereafter. In addition, $\|\vartheta_\kappa\| \geq \Omega_\kappa$ is equivalent to $|s_\kappa| \geq \Omega_\kappa^4$. Hence, the update laws just need to be activated when $|s_\kappa| \geq \lambda_\kappa$, with λ_κ chosen in the set $C = \left\{0 < \lambda_\kappa \leq \Omega_\kappa^4\right\}$. In this way, the adaptation action is always feasible without introducing any unwanted effects on the control performance.

In addition, letting $\gamma_\kappa = \left[|s_\kappa|^{\frac{-1}{4}} \boldsymbol{z}_\kappa^T \ |s_\kappa|^{\frac{-1}{4}} \bar{\varepsilon}_\kappa \ \chi_\kappa\right]^T$, (30) becomes:

$$\dot{V}_\kappa \leq \gamma_\kappa^T \begin{bmatrix} \boldsymbol{K}_\kappa & \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa & \boldsymbol{O}_{2\times 1} \\ \boldsymbol{B}_\kappa^T \boldsymbol{P}_\kappa & -\gamma_\kappa^2 & 0 \\ \boldsymbol{O}_{1\times 2} & 0 & -k_\kappa \end{bmatrix} \gamma_\kappa + |s_\kappa|^{\frac{-1}{2}} \gamma_\kappa^2 \bar{\varepsilon}_\kappa^2 + \chi_\kappa \Delta u_\kappa \tag{32}$$

Applying Young's inequality for $\chi_\kappa \Delta u_\kappa$, one can have:

$$\dot{V}_\kappa \leq \gamma_\kappa^T \begin{bmatrix} \boldsymbol{K}_\kappa & \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa & \boldsymbol{O}_{2\times 1} \\ \boldsymbol{B}_\kappa^T \boldsymbol{P}_\kappa & -\gamma_\kappa^2 & 0 \\ \boldsymbol{O}_{1\times 2} & 0 & 1/\left(2\gamma_\kappa^2\right) - k_\kappa \end{bmatrix} \gamma_\kappa + |s_\kappa|^{\frac{-1}{2}} \gamma_\kappa^2 \left(\bar{\varepsilon}_\kappa^2 + \frac{1}{2}|s_\kappa|^{\frac{1}{2}} \Delta u_\kappa^2\right) \tag{33}$$

and, let $\mathrm{Y}_\kappa = \dot{V}_\kappa + |s_\kappa|^{\frac{-1}{2}} \left(|s_\kappa| - \gamma_\kappa^2 \bar{\varepsilon}_\kappa^2 - 2^{-1}|s_\kappa|^{\frac{1}{2}} \gamma_\kappa^2 \Delta u_\kappa^2\right)$, one gets:

$$\mathrm{Y}_\kappa \leq \gamma_\kappa^T \boldsymbol{N}_\kappa \gamma_\kappa, \ \ \boldsymbol{N}_\kappa = \begin{bmatrix} \boldsymbol{K}_\kappa + \boldsymbol{C}_\kappa^T \boldsymbol{C}_\kappa & \boldsymbol{P}_\kappa \boldsymbol{B}_\kappa & \boldsymbol{O}_{2\times 1} \\ \boldsymbol{B}_\kappa^T \boldsymbol{P}_\kappa & -\gamma_\kappa^2 & 0 \\ \boldsymbol{O}_{1\times 2} & 0 & 1/\left(2\gamma_\kappa^2\right) - k_\kappa \end{bmatrix} \tag{34}$$

where $\boldsymbol{C}_\kappa = [1 \ \ 0]$. One can see that if $\mathrm{Y}_\kappa \leq 0$, it implies $|s_\kappa| - |s_\kappa|^{\frac{1}{2}} \gamma_\kappa^2 \Delta u_\kappa^2/2 - \gamma_\kappa^2 \bar{\varepsilon}_\kappa^2 \leq 0$.

Solving this with respect to $|s_\kappa|^{\frac{1}{2}}$, one gets $s_\kappa \in \{r_\kappa - \Phi_\kappa/4 \leq |s_\kappa| \leq r_\kappa + \Phi_\kappa/4\}$ where $r_\kappa = \gamma_\kappa^4 \Delta u_{\kappa_{\max}}^4/16$ and $\Phi_\kappa = \gamma_\kappa^2 \left(\gamma_\kappa^2 \Delta u_{\kappa_{\max}}^4 + 16\bar{\varepsilon}_{\kappa_{\max}}^4\right)/4$. Thus, the upper bound of the sliding surface s_κ is defined as:

$$|s_\kappa| \leq \frac{1}{16} \gamma_\kappa^2 \left(2\Delta u_{\kappa_{\max}}^4 \gamma_\kappa^2 + 16\bar{\varepsilon}_{\kappa_{\max}}^2 + 4\Delta u_{\kappa_{\max}}^2 \gamma_\kappa \sqrt{\frac{1}{4}\Delta u_{\kappa_{\max}}^4 \gamma_\kappa^2 + 4\bar{\varepsilon}_{\kappa_{\max}}^2}\right) \tag{35}$$

Additionally, letting $\boldsymbol{H}_\kappa = -\boldsymbol{N}_\kappa$, $\mathrm{Y}_\kappa \leq 0$ if and only if $\boldsymbol{H}_\kappa > 0$. Hence, let us transform $\boldsymbol{H}_\kappa$ following the Gaussian elimination method, that is:

$$\bar{\boldsymbol{H}}_\kappa = \begin{bmatrix} n_{3\kappa} n_{4\kappa} - 1 & 0 & n_{4\kappa} & 0 \\ 0 & n_{3\kappa}/2 & -n_{3\kappa}/4 & 0 \\ 0 & 0 & p_\kappa & 0 \\ 0 & 0 & 0 & k_\kappa - 1/(2\gamma_\kappa^2) \end{bmatrix} \tag{36}$$

where $p_\kappa = \gamma_\kappa^2 - n_{4\kappa}^2 (n_{3\kappa} n_{4\kappa} - 1) - n_{3\kappa}/8$. One can see that $\boldsymbol{H}_\kappa > 0$ if all pivots in (36) are positive.

3.5. Gain Tuning

In view of (35) and (36), the minimal upper bound of s_κ can be achieved by minimizing γ_κ^2 over $n_{3\kappa}$, subjects to $n_{3\kappa} n_{4\kappa} - 1 > 0$, $n_{3\kappa}/2 > 0$, $p_\kappa > 0$, and $k_\kappa - 1/\left(2\gamma_\kappa^2\right) > 0$. Recalling $n_{3\kappa}^2 = 4n_{4\kappa}$, the first constraint becomes $n_{3\kappa} > \sqrt[3]{4}$. This condition also satisfies the second constraint. Moreover, substituting $n_{3\kappa} > \sqrt[3]{4}$ to p_κ, the third constraint becomes $\zeta_\kappa - m_\kappa - \sqrt[3]{4}/8 > 0$ with $\zeta_\kappa = \gamma_\kappa^2$ and $m_\kappa = n_{3\kappa}^4/\left(4n_{3\kappa}^3 - 16\right)$. Assuming we can find ζ_κ and m_κ to satisfy the third constraint, then γ_κ and $n_{3\kappa}$ can be found from: $\gamma_\kappa = \sqrt{\zeta_\kappa}$ and $n_{3\kappa}^4/\left(4n_{3\kappa}^3 - 16\right) - m_\kappa = 0$, respectively. To ensure the existence of $n_{3\kappa}$ and γ_κ, m_κ and ζ_κ have to satisfy the following conditions $m_\kappa > \sqrt[3]{16/27}$ and $\zeta_\kappa > 0$. Besides, substituting

$\zeta_\kappa = \gamma_\kappa^2$ to the fourth constraint, it turns out that $\zeta_\kappa > 1/(2k_\kappa)$. This also satisfies $\zeta_\kappa > 0$. Thereby, the minimization problem can be stated as:

$$\begin{aligned} &\min_{m_\kappa,\, \zeta_\kappa} \quad \zeta_\kappa \\ &s.t. \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \zeta_\kappa - \begin{bmatrix} \frac{\sqrt[3]{4}}{8} & 0 \\ 0 & \frac{1}{2k_\kappa} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} m_\kappa > 0,\ m_\kappa - \sqrt[3]{\frac{16}{27}} > 0 \end{aligned} \tag{37}$$

The problem in (37) is an EVP and can be approximated in an unconstrained form using the log-barrier function [27]. Moreover, let ℓ_κ^* denotes the optimal value of ζ_κ, so for each $\ell_\kappa > \ell_\kappa^*$, the constraints in (37) and $\zeta_\kappa < \ell_\kappa$ are feasible. Assume that these constraints have a bounded feasible set, and the analytic center $\zeta_\kappa^*(\ell_\kappa)$ is defined as:

$$\begin{aligned} \zeta_\kappa^*(\ell_\kappa) &= \underset{\zeta_\kappa,\, m_\kappa}{\operatorname{argmin}} \xi_\kappa(\zeta_\kappa,\ m_\kappa), \\ \xi_\kappa(\zeta_\kappa,\ m_\kappa) &= -\log(\zeta_\kappa - k_\kappa/2) - \log\left(\zeta_\kappa - m_\kappa - \sqrt[3]{4}/8\right) \\ &- \log\left(m_\kappa - \sqrt[3]{16/27}\right) - \log(\ell_\kappa - \zeta_\kappa) \end{aligned} \tag{38}$$

Given the feasible starting values, $\zeta_\kappa^{(0)}$, $m_\kappa^{(0)}$, and $\ell_\kappa^{(0)}$, the problem in (38) can be solved by the method of centers, whose pseudo-algorithm can be described as follows (where ι_κ and ϑ_κ represent of stopping criteria, ν_κ is a gain with $0 < \nu_\kappa < 1$, α is the learning rate of the inner loop):

4. Experimental Studies

4.1. Implementation

In experiments, the prototype of the water-powered aerial robot is placed on the test bench such that the longitudinal, lateral, and heading motions are eliminated, as shown in Figure 4. Moreover, the detailed system parameters of the prototype were fully described in [7].

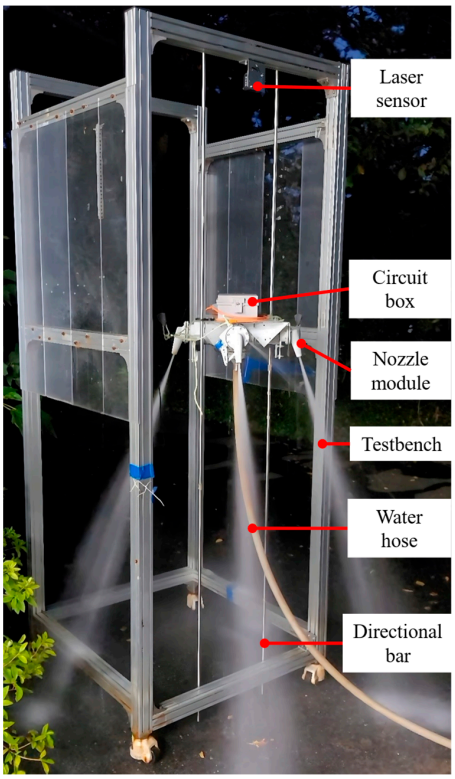

Figure 4. Prototype of aerial robot placed on test bench.

The parameters in Algorithm 1 are chosen as: $\alpha = 0.001$, $\zeta_\kappa^{(0)} = 90$, $m_\kappa^{(0)} = 20$, $\ell_\kappa^{(0)} = 100$, $k_\kappa = 45$, $\iota_\kappa = 0.04$, $\vartheta_\kappa = 0.01$, $\vartheta_\kappa = 0.99$. This algorithm is then implemented in MATLAB R2024a software. After 25,083 outer loop iterations under 0.298119 [s], ζ_κ and m_κ converge to 1.0815 and 0.84154, respectively. Hence, the controller's gains are: $\boldsymbol{n}_1 = diag\{0.8,\ 10,\ 10\}$, $\boldsymbol{n}_2 = diag\{18,\ 55,\ 55\}$, $n_{3\kappa} = 2.6$, $n_{4\kappa} = 1.7$, $\upsilon_z = 0.1$, $\lambda_\kappa = 0.1$, $\eta_\kappa = \mu_\kappa = 0.001$, $\upsilon_\phi = \upsilon_\theta = 0.01$. The initial network's parameters are: $j = 5$, $\hat{\beta}_\kappa = 5$, $\hat{\boldsymbol{w}}_z = 10^{-4}\mathbf{1}_{1\times5}$, $\hat{\boldsymbol{\sigma}}_z = [-2\ \ -6\ \ 0\ \ 2\ \ 6]$, $\hat{\boldsymbol{\sigma}}_\phi = \hat{\boldsymbol{\sigma}}_\theta = [-3\ \ -5\ \ 0\ \ 3\ \ 5]$, $\hat{\boldsymbol{w}}_\phi = \hat{\boldsymbol{w}}_\theta = 10^{-2}\mathbf{1}_{1\times5}$, where $\mathbf{1}_{1\times5}$ is the 1-by-5 matrix of 1.

Algorithm 1: The method of centers

given $\zeta_\kappa^{(0)}, m_\kappa^{(0)}, \ell_\kappa^{(0)}, \nu_\kappa, \iota_\kappa, \vartheta_\kappa\ \alpha,\ j = 0,$ and $i = 0$
repeat until $\ell_\kappa^{(i)} - \zeta_\kappa^{(i)} < \iota_\kappa$**:**
1. $\ell_\kappa^{(i+1)} := (1-\nu_\kappa)\gamma_\kappa^{(i)} + \nu_\kappa \ell_\kappa^{(i)}$
2. $j = 0$
for $\ell_\kappa^{(i+1)} := (1-\nu_\kappa)\gamma_\kappa^{(i)} + \nu_\kappa \ell_\kappa^{(i)}$ **repeat until** $\left\| \left[\partial \xi_\kappa / \partial \zeta_\kappa \quad \partial \xi_\kappa / \partial m_\kappa \right] \right\| < \vartheta_\kappa$**:**
1. $\zeta_\kappa^{(j+1)} = \zeta_\kappa^{(j)} - \alpha \nabla_{\zeta_\kappa} \xi_\kappa$
2. $m_\kappa^{(j+1)} = m_\kappa^{(j)} - \alpha \nabla_{m_\kappa} \xi_\kappa$
3. $j = j + 1$
end
3. $\gamma_\kappa^{(i+1)} := \gamma_\kappa^* \left(\ell_\kappa^{(i+1)} \right)$
4. $i = i + 1$
end

The experimental setup is shown in Figure 5. Specifically, the proposed controller is implemented on a Portenta H7 microcontroller (Arduino SRL, Ivrea, Italy) with a clock speed of 480 [MHz]. This microcontroller can ensure a sample of 0.01 [s] for the entire system. In addition, data are collected using MATLAB R2024a software on a Ryzen 5500U laptop (Acer Inc., Taipei, China) via an HC-05 Bluetooth module. The system's baud rate is set to 921.6 [Kbps]. In addition, the altitude position and angular position of the robot are measured using the DT50-2 (SICK AG, Waldkirch, Baden-Wurttemberg, Germany) and Mti-680G sensors (Xsens Technologies B.V., Enschede, The Netherlands), respectively.

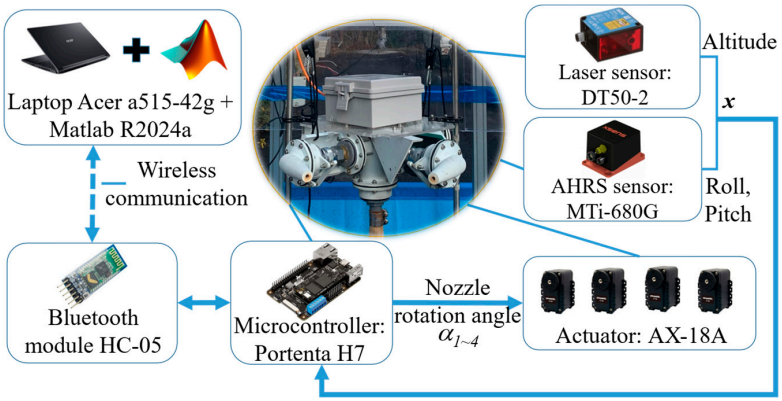

Figure 5. Block diagram of experimental setup.

4.2. Experimental Results

The altitude of the robot follows the desired step-like reference with a peak-to-peak amplitude of 0.8 [m], as described in [7], while its roll and pitch angles are stabilized at the origin. This scenario is conducted in 75 [s], consisting of takeoff (5–15 [s]), climb (30–40 [s]),

hovering (15–30 and 40–55 [s]), and landing (55–75 [s]) processes. The PID controller [7], conventional SMC, and AWSTC in (7) are deployed for comparison.

The tracking errors are shown in Figure 6. Overall, all controllers can provide stability to the system. Particularly, the PID controller exhibits a poor transient response. The comparison SMC can provide good tracking performance thanks to its discontinuous input. However, this also causes chattering, especially during takeoff and landing processes, where high overshoots are also captured. The latter AWSTC, known as a second-order continuous sliding mode control algorithm, provides better performance compared to the PID controller and SMC. Consequently, overshoots and steady-state errors are significantly reduced. However, the influence of the water hose is still significant, especially during the landing process. Finally, this influence can be effectively compensated by the proposed controller, thanks to the neural network, leading to better performance compared to the three controllers mentioned.

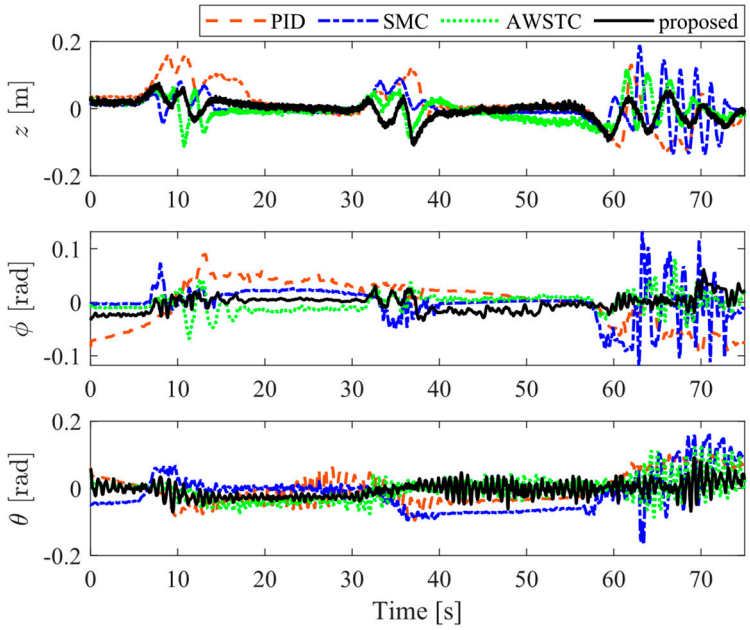

Figure 6. Tracking errors.

The control efforts are illustrated in Figure 7. Overall, all nozzles consistently remained within the available operating range during flight. However, the second, third, and fourth nozzles controlled by the PID controller and SMC reach saturation. At this point, the windup effect is produced due to the integral action. However, in the case of AWSTC and the proposed controller, this effect is prevented by the anti-windup compensator. Hence, the nozzles' saturation in these two cases is significantly reduced. Moreover, the effectiveness of the proposed system can be quantified using numerical performance metrics, i.e., root mean square error (RMSE) and integral of the square value (ISV), as shown in Table 1. Based on these two performance metrics, i.e., RMSE and ISV, one can first conclude that both the AWSTC and proposed controller outperform the comparative PID and conventional sliding controllers in terms of stability and tracking accuracy. In particular, the ISV index reflects the integral of the nozzle angles over time, rather than the actual actuator energy consumption. In the case of PID and SMC controllers, when the nozzle angle reaches the actuator's upper saturation limit, these controllers continue to accumulate these saturated angle values, leading to a higher ISV. This issue is handled in both the AWSTC and the proposed controller through the use of anti-windup schemes, which prevent excessive accumulation beyond saturation. The fact that the proposed controller shows a larger ISV compared to the AWSTC does not imply higher energy consumption or greater control

effort. Instead, it suggests that the proposed controller exhibits a different rotational behavior, which results in a distinct nozzle angle trajectory.

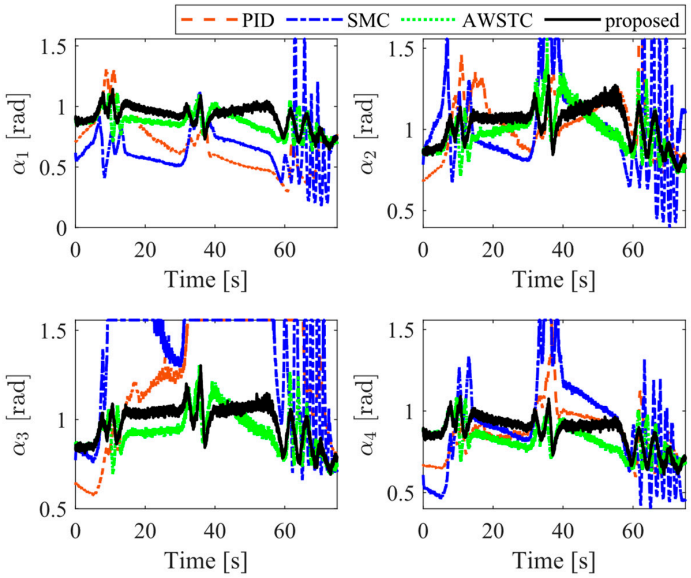

Figure 7. Control efforts.

Table 1. Performance metrics.

		PID	SMC	AWSTC	Proposed
RMSE	z [m]	0.056	0.039	0.032	0.027
	ϕ [rad]	2.516	1.699	0.928	0.870
	θ [rad]	2.823	3.471	2.092	1.367
ISV [rad^2]		317.166	338.214	261.417	287.402

In the RBF neural network, we select the third weight and third expectation value, along with the standard deviation, as representative adaptation parameters of the neural network for each motion κ, as shown in Figure 8. These parameters are updated following the sliding surface, e.g., $\hat{\omega}_{z3}$, $\hat{\sigma}_{z3}$, $\hat{\beta}_z$ are updated during takeoff and landing, where $|s_z| > \lambda_z = 0.1$. Consequently, the influences are estimated effectively based on the updates of the parameters in the RBF neural network.

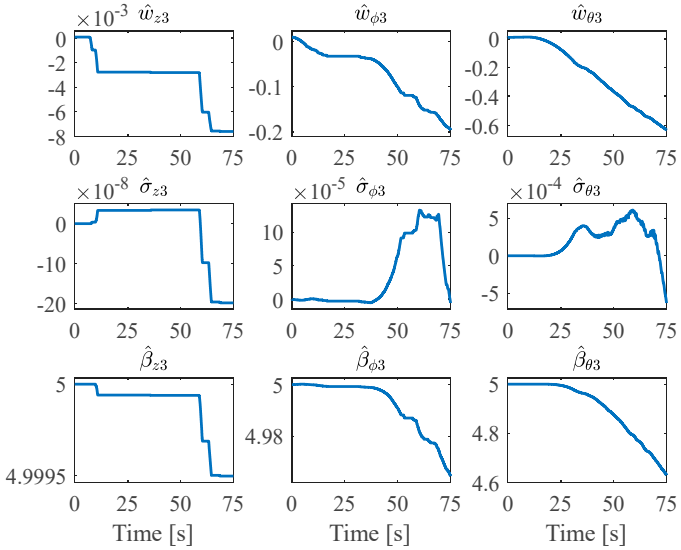

Figure 8. Adaptation of the parameters in the neural network.

5. Conclusions

This article has proposed the design of an intelligent robust controller for the 3-DoF aerial robots powered by water. The proposed controller consists of two parts: (1) the AW-STC, providing stability to the system under actuator saturation; and (2) the fully adaptive RBF neural network, compensating for unexpected influences. The proposed system is stable in the sense of Lyapunov stability theory. The problem of control parameter optimization was formulated as an EVP and solved using the method of centers. Experimental results demonstrated that the proposed system exhibits superior tracking performance and strong robustness against unexpected influences, especially the water hose vibration. Additionally, overall actuator saturation was significantly reduced.

At present, the water hose is not considered, and the nozzle rotation range is limited. Hence, future work will focus on: (1) mathematically investigating the water hose to improve control performance; (2) modifying the servo drive mechanism for unlimited nozzle rotation; and (3) studying the full 6-DoF motion control of the robot.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/act14040197/s1, Video S1: Intelligent Robust Motion Control of Aerial Robot.

Author Contributions: Conceptualization, Y.-B.K.; methodology, Y.-B.K. and T.H.; software, C.-T.D. and T.-D.N.; validation, C.-T.D., T.-D.N. and J.-S.P.; formal analysis, C.-T.D. and J.-S.P.; investigation, C.-T.D., T.H. and J.-S.P.; resources, J.-S.P.; data curation, C.-T.D.; writing—original draft preparation, C.-T.D. and J.-S.P.; writing—review and editing, C.-T.D. and T.H.; visualization, C.-T.D.; supervision, Y.-B.K.; project administration, Y.-B.K.; funding acquisition, Y.-B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Research Foundation (NRF), South Korea, under Project BK21 FOUR (Smart Robot Convergence and Application Education Research Center); and in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2022R1A2C1003486).

Data Availability Statement: The original contributions presented in this study are included in the article/Supplementary Materials. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Liu, X.; Zhou, H. Unmanned Water-Powered Aerial Vehicles: Theory and Experiments. *IEEE Access* **2019**, *7*, 15349–15356. [CrossRef]
2. Ando, H.; Ambe, Y.; Ishii, A.; Konyo, M.; Tadakuma, K.; Maruyama, S.; Tadokoro, S. Aerial Hose Type Robot by Water Jet for Fire Fighting. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1128–1135. [CrossRef]
3. Yamaguchi, T.; Ambe, Y.; Ando, H.; Konyo, M.; Tadakuma, K.; Maruyama, S.; Tadokoro, S. A Mechanical Approach to Suppress the Oscillation of a Long Continuum Robot Flying with Water Jets. *IEEE Robot. Autom. Lett.* **2019**, *4*, 4346–4353. [CrossRef]
4. Yamauchi, Y.; Ambe, Y.; Konyo, M.; Tadakuma, K.; Tadokoro, S. Passive Orientation Control of Nozzle Unit with Multiple Water Jets to Expand the Net Force Direction Range for Aerial Hose-Type Robots. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5634–5641. [CrossRef]
5. Yamauchi, Y.; Ambe, Y.; Konyo, M.; Tadakuma, K.; Tadokoro, S. Realizing Large Shape Deformations of a Flying Continuum Robot with a Passive Rotating Nozzle Unit That Enlarges Jet Directions in Three-Dimensional Space. *IEEE Access* **2022**, *10*, 37646–37657. [CrossRef]
6. Ambe, Y.; Yamauchi, Y.; Konyo, M.; Tadakuma, K.; Tadokoro, S. Stabilized Controller for Jet Actuated Cantilevered Pipe Using Damping Effect of an Internal Flowing Fluid. *IEEE Access* **2022**, *10*, 5238–5249. [CrossRef]
7. Dinh, C.-T.; Kim, Y.-B.; Huynh, T.; Lee, D.-H. Experimental Study on Altitude Motion Control of Unmanned Waterpowered Aerial Vehicle Using Nozzle Rotation Mechanism. *J. Korean Soc. Precis. Eng.* **2024**, *41*, 789–796. [CrossRef]

8. Lee, S.; Son, H. Antisway Control of a Multirotor with Cable-Suspended Payload. *IEEE Trans. Control Syst. Technol.* **2021**, *29*, 2630–2638. [CrossRef]
9. Akhtar, A.; Saleem, S.; Shan, J. Path Invariant Controllers for a Quadrotor with a Cable-Suspended Payload Using a Global Parameterization. *IEEE Trans. Control Syst. Technol.* **2022**, *30*, 2002–2017. [CrossRef]
10. Goodman, J.R.; Beckers, T.; Colombo, L.J. Geometric Control for Load Transportation with Quadrotor UAVs by Elastic Cables. *IEEE Trans. Control Syst. Technol.* **2023**, *31*, 2848–2862. [CrossRef]
11. Costantini, E.; de Angelis, E.L.; Giulietti, F. Cooperative Drone Transportation of a Cable-Suspended Load: Dynamics and Control. *Drones* **2024**, *8*, 434. [CrossRef]
12. Luo, C.; Du, Z.; Yu, L. Neural Network Control Design for an Unmanned Aerial Vehicle with a Suspended Payload. *Electronics* **2019**, *8*, 931. [CrossRef]
13. Maezawa, Y.; Ambe, Y.; Yamauchi, Y.; Konyo, M.; Tadakuma, K.; Tadokoro, S. Translational Disturbance Rejection for Jet-Actuated Flying Continuum Robots on Mobile Bases. *IEEE Robot. Autom. Lett.* **2023**, *8*, 7456–7463. [CrossRef]
14. Dinh, C.-T.; Huynh, T.; Lee, D.-H.; Kim, Y.-B. Design and Experiments of Angular Motion Control for a Novel Unmanned Water-Powered Aerial Vehicle. *IEEE Access* **2023**, *11*, 112719–112730. [CrossRef]
15. Lee, D.-H.; Huynh, T.; Kim, Y.-B.; Soumayya, C. Motion control system design for a flying-type firefighting system with water jet actuators. *Actuators* **2021**, *10*, 275. [CrossRef]
16. Huynh, T.; Kim, Y.-B. Motion control system design for a novel waterpowered aerial system for firefighting with flow-regulating actuators. *Drones* **2023**, *7*, 162. [CrossRef]
17. Liu, Z.; Liu, J.; Zhang, O.; Zhao, Y.; Chen, W.; Gao, Y. Adaptive Disturbance Observer-Based Fixed-Time Tracking Control for Uncertain Robotic Systems. *IEEE Trans. Ind. Electron.* **2024**, *71*, 14823–14831. [CrossRef]
18. Liu, Z.; Zhang, O.; Gao, Y.; Zhao, Y.; Sun, Y.; Liu, J. Adaptive Neural Network-Based Fixed-Time Control for Trajectory Tracking of Robotic Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *70*, 241–245. [CrossRef]
19. Tao, X.; Liu, K.; Yang, J.; Chen, Y.; Chen, J.; Zhu, H. Sliding Mode Backstepping Control of Excavator Bucket Trajectory Synovial in Particle Swarm Optimization Algorithm and Neural Network Disturbance Observer. *Actuators* **2025**, *14*, 9. [CrossRef]
20. Zhuo, H.; Yang, Z.; You, Y.; Xu, N.; Liao, L.; Wu, J.; He, J. A Hierarchical Control Method for Trajectory Tracking of Aerial Manipulators Arms. *Actuators* **2024**, *13*, 333. [CrossRef]
21. Li, T.; Xiao, T. Physics-Informed Neural Network-Based Input Shaping for Vibration Suppression of Flexible Single-Link Robots. *Actuators* **2025**, *14*, 14. [CrossRef]
22. Zhang, W.; Yu, B. Adaptive Predefined Time Control for Strict-Feedback Systems with Actuator Quantization. *Actuators* **2024**, *13*, 366. [CrossRef]
23. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
24. Montoya-Cháirez, J.; Rossomando, F.G.; Carelli, R.; Santibáñez, V.; Moreno-Valenzuela, J. Adaptive RBF neural network-based control of an underactuated control moment gyroscope. *Neural Comput. Appl.* **2021**, *12*, 6805–6818. [CrossRef]
25. Filippov, A.F. *Differential Equations with Discontinuous Right-Hand Side*; Kluwer: Dordrecht, The Netherlands, 1988.
26. Moreno, J.A.; Osorio, M. Strict Lyapunov Functions for the Super-Twisting Algorithm. *IEEE Trans. Autom. Control* **2012**, *57*, 1035–1040. [CrossRef]
27. Boyd, S.; El Ghaoui, L.; Feron, E.; Balakrishnan, V. *Linear Matrix Inequalities Syst. Control Theory*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1994.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.