applied sciences

MDPI

Article

Robust Adaptive Finite-Time Synergetic Tracking Control of Delta Robot Based on Radial Basis Function Neural Networks

Phu-Cuong Pham [1] and Yong-Lin Kuo [1,2,*]

[1] Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei 106335, Taiwan
[2] Center of Automation and Control, National Taiwan University of Science and Technology, Taipei 106335, Taiwan
* Correspondence: kuo@mail.ntust.edu.tw

Abstract: This paper presents a robust proportional derivative adaptive nonsingular finite-time synergetic tracking control (PDAFS) for a parallel Delta robot system. First, a finite-time synergetic controller combined with a proportional derivative (PD) control is constructed based on an object-oriented model to fulfill the robust tracking control of the robot. Then, an adaptive radial basis function approximation neural network (RBF) is designed to compensate for the effects of uncertainty parameters and external disturbances. Second, a second-order sliding mode (SOSM) differentiator is implemented to reduce the chattering noises due to the low-resolution encoders. Third, the stability theorems of the proposed control scheme are provided, where the Lyapunov stability theory is used to prove the theorems. Then, simulations of the helix trajectory tracking and the pick-and-place task are demonstrated on the Delta robot to validate the advantages of the proposed control scheme. Based on the advances, an implementing control system of the proposed controller is performed to improve the Delta robot's performance in the experiments.

Keywords: adaptive synergetic control; dynamics delta robot; radial basis function approximation; second-order sliding mode observer

check for updates

Citation: Pham, P.-C.; Kuo, Y.-L. Robust Adaptive Finite-Time Synergetic Tracking Control of Delta Robot Based on Radial Basis Function Neural Networks. *Appl. Sci.* **2022**, *12*, 10861. https://doi.org/10.3390/app122110861

Academic Editors: Mingcong Deng, Augusto Ferrante, Mihaiela Iliescu and Tai Yang

Received: 18 September 2022
Accepted: 22 October 2022
Published: 26 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 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

Delta parallel robot is the most successful commercial parallel robot to date. The parallel robot has been employed for high-precision tracking and high-speed motions to boost output [1,2]. The robot has three kinematics branches consisting of two arms, and the branches connect a base and a moving plate. Due to the complexity of the closed-loop structure and kinematic constraints of the robot, the robot model is a multi-variable, multi-parameter, coupling, and multi-degree-of-freedom nonlinear system. Many studies have shown the significance of kinematic [3–5] and dynamic models [6–9] on path planning and tracking precision. Accordingly, numerous model-based control techniques are employed to improve the performance of parallel robots. For instance, Codourey proposed a proportional derivative feedforward (PDFF) controller based on a dynamic model of the Delta robot in [10,11] to give more precise trajectory tracking than a kinematic controller. In a recent work [12], a dynamic feedforward decoupling control scheme was proposed for a compensated feedforward gravity and an independently adjustable three PID control loop corresponding to the position, velocity, and current controls of a four-degrees-of-freedom parallel robot. Although the simulation results reveal that the system can reduce tracking errors, it is difficult to be applied due to the decoupling factors in serial control loops, which depend not only on the correctness of the dynamic modal transformations but also on the noises of the feedback signal. In addition, the two model-based methods require not only the accuracy of parameters but also lack influences of uncertainty parameters and unknown disturbances in the systems.

To solve robust tracking problems and reduce the influence of external disturbance, a sliding mode control (SM) technique based on the Delta robot dynamics with uncertain parameters was proposed in [13]. This approach guarantees stability by keeping system states on a sliding plane. Another SM improvement by a terminal attractor technique is the so-called terminal sliding mode (TSM) control, proposed in [14] for finite-time stabilization of robotic manipulators. However, both SM and TSM techniques have a drawback, which produces high-frequency oscillations of the controller outputs, known as chattering noises. The noises can impose vibration on the mechanical actuators. To eliminate the chattering phenomenon, another control methodology for modern and complex systems led to the introduction of the synergetic control (SC) theory [15] by Kolesnikov. Similar to SM properties, the advantages of synergetic control are well-suited for order reductions of the controlled system, robustness to external disturbance, and better control of the off-manifold dynamics. In synergetic control papers [16–18], synergetic-based controllers generate chattering-free dynamics of nonlinear systems. Inspired by the TSM, finite-time synergetic control (FS) schemes to control serial robot manipulators were proposed in [19,20]. The finite-time synergetic control (FS) scheme employed the synergetic theory and a terminal attractor technique to control serial robot manipulators. The new synergetic approach is allowed to perform well in multiple connected nonlinear systems with continuous, discontinuous, discrete-time, terminal, and adaptive control, such as robots [20,21], electric drives [22], and flying apparatus [23]. Based on those studies, the benefits of a synergetic approach, such as robustness, quick dynamic reactions, and chattering-free phenomena, are well suited for the Delta robot.

In addition, the Delta parallel robot is a highly nonlinear complexity system. Therefore, developing a suitable control scheme for the robot with parametric and non-parametric uncertainties and external disturbance represents a significant challenge. Boudjedir and co-workers proposed a nonlinear controller to solve robust and precise trajectory problems using SM or a nonlinear PD controller based on iterative learning [24–26]. However, the schemes must require a long time to follow continuous repetitive or non-repetitive trajectories before improving the tracking errors. To online estimate uncertain plant parameters, different kinds of adaptive control strategies have been performed to deal with the uncertain parameter or disturbance. The work in [27] proposed an adaptation law for dynamics and friction compensation of the dynamic model of the 2-DOF parallel mechanism based on a Lyapunov method. In ref. [28], an online estimation was proposed for simulating trajectory tracking control with nonlinear joint friction models, which were compensated to the nominal Delta model. Another adaptive control inside an active disturbance rejection framework used the least mean square technique in [29] to produce a reliable tracking control for the low-speed Delta platform, where the researchers did not consider the joint velocity data of the system. As can be seen, uncertainty parameters in the adaptive controllers must be analyzed explicitly in a specific model, which represents the effects of the system tracking errors.

In recent years, there have been numerous adaptive intelligent control techniques. On the one hand, a systematic analysis with sensory noise and uncertainty robustness for a type 2 fuzzy logic was used to position the control of the end-effector of the parallel Delta robot [30]. Or a utilization of fuzzy logic systems can estimate the unknown nonlinear behavior of simple examples and adapt successfully via synergetic control theory [31]. However, the linguistic rules and optimal dataset for stability guarantee remain challenging issues. On the other hand, uncertainty parameters in a model are estimated by approximations using neural networks (NNs). The study in [32] is an example of using NNs to create compensators for the dynamic model. In particular, the robust integral of the sign error (RISE) controller with a B-spline NN feedforward compensation was applied to a Delta robot to regulate the trajectory tracking for a pick-place application. An intelligent compensation term added to the system control can reduce the tracking error considerably and cancel the steady-state error. However, only simulating validations of these NNs are taken into account. Based on a linear motor model with unknown frictions [33], a

processing structure of two-layer NNs was proposed successfully to compensate for the effects of the friction model in both simulations and experiments. Ref. [34] gave an output feedback controller with a feedforward term based on a radial basis function (RBF) neural network that dealt with uncertainty compensation in a robotic exoskeleton. It is shown that an RBF network may approximate a continuously nonlinear function with benefits such as low-cost computation and quick convergence. In this work, RBF networks are utilized in addition to a robust synergetic controller to estimate unknown parameters for control quality enhancement.

Based on the benefits and limitations of the above literature review, a novel combination of the advantages of a proportional derivative, a robust synergetic, and an RBF network controller based on the object-oriented technique [7] in a Delta robot system with the generalized disturbance is proposed. Therefore, the main contributions of this paper lie in the following aspects:

1. A systematic controller using a PD controller with a finite-time synergetic technique is developed based on the Delta robot dynamics for robust trajectory tracking control. In the proposed controller structure, unknown parameters and external disturbance of the DC motors and the robot manipulator are analyzed in a generalized disturbance model, then compensated torques are estimated online through RBF 2-layer neural networks.
2. The stability of the proposed control framework is analyzed and proven through the Lyapunov stability theory.
3. The features of the proposed controller are successfully applied to the parallel Delta robot in simulation and practical experiments with different trajectory examples. A comparative study between PD, feedforward, SM, and the proposed approach is provided to reveal the advantages and improvements of the latter.

The rest of this paper is organized as follows. A nonlinear model of the Delta robot with DC motors is analyzed in Section 2. Then, the robust adaptive synergetic control based on RBF neural networks is proposed in Section 3. Finally, the simulation results in Section 4 and the experimental results in Section 5 present the advantages of the proposed method in comparison with the results obtained by the other controllers.

2. Nonlinear Dynamics of Delta Robot Model

2.1. DC Motor Servo Model

The DC motor model for the robot system is described by electrical and mechanical differential equations [35,36]:

$$u_a = R_a i_a + L_a \frac{di_a}{dt} + K_e \dot{q} \tag{1}$$

$$K_t i_a = \tau_m = J_m \ddot{q} + b\dot{q} + \tau_L \tag{2}$$

where u_a is voltage input applied to the motor's armature, R_a is armature resistance, L_a is armature inductance, i_a is armature current, q and $\dot{q}$ are angular displacement and velocity of the motor shaft, the product τ_m is torque generated by armature current with torque constant K_t, b represents the coefficient of velocity-dependent friction in the motor, and τ_L is torque of load, back electromotive force (EMF) coefficient is denoted by K_e. *The dynamic parameters of the DC motor servo in Table 1 are required by performing a system identification experiment* [36] *and using the MATLAB toolbox* [37].

2.2. Delta Robot Manipulator

Consider the experimental Delta robot and its parameters illustrated in Figure 1 and Table 2, which consists of three symmetric branches constrained in a kinematic manner by the universal or spherical joint at the traveling plate. The complexity of the model arises mainly due to the movement of the forearms in branches. The forearm is constructed of two lightweight aluminum rods and attached rod end bearings at their extremities. The

rod end bearings of the forearms transfer the motion of the upper arms to the traveling plate, giving it three degrees of freedom. By eliminating the forearms' rotational inertia, we may reduce the intricacy of their motion. This assumption is not very restricting due to the weight distribution of the forearm to the extremities. Moreover, a generalized disturbance of the Delta robot is analyzed in Section 3, so we can neglect the effects of joint friction and elasticity in this section to simplify the dynamic model of the Delta manipulator.

Table 1. Parameters of the DC motor.

Symbol	Description	Units	Values
L_a	armature inductance	H	1.8×10^{-4}
R_a	armature resistance	Ω	3.4
K_t	torque constant	Nm/A	1
K_e	back EMF coefficient	Nm/A	1
b	friction coefficient	Nm	0.01859
J_m	motor inertia	kg m^2	0.0443

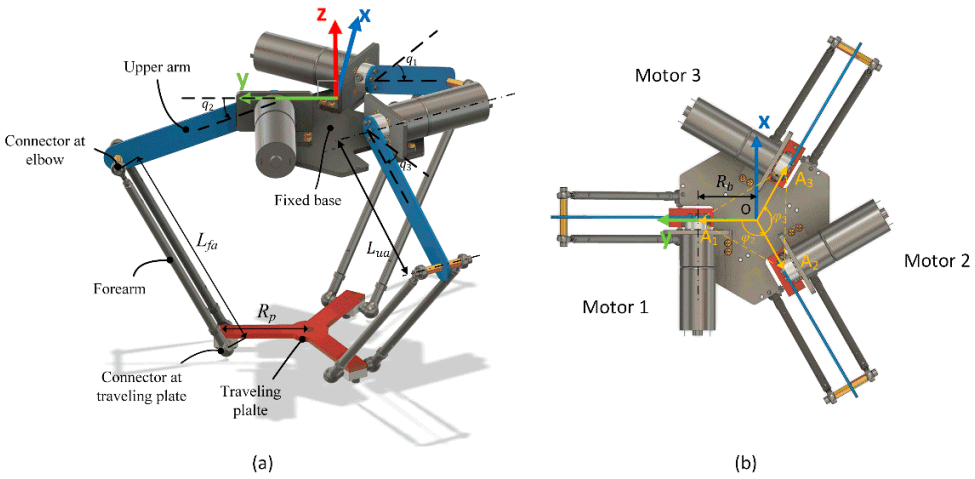


Figure 1. The Delta robot in 3D view (**a**) and top-view (**b**).

Table 2. Kinematic and dynamic parameters of Delta manipulator.

Symbol	Description	Units	Values
L_{ua}	Length of upper arm	m	0.175
L_{fa}	Length of forearm	m	0.25
R_p	Radius of traveling plate	m	0.09
R_b	Radius of fixed base	m	0.07
m_{ua}	Mass of upper arm	kg	0.046
m_{fa}	Mass of forearm	kg	0.082
m_c	Mass of connector	kg	0.048
m_p	Mass of traveling plate	kg	0.317

The inverse dynamic model of the Delta robot is developed as [10,11] based on the Newton–Euler method with the following simplifying hypotheses:

- The rotational inertia of forearms is neglected.
- For analytical purposes, the masses of forearms are optimally separated into portions and at their extremities: a two-thirds majority part at its upper extremity and the other part at its lower extremity, which contributes to the traveling plate mass.
- Friction effects and elasticity are neglected.

With the above-mentioned simplifying hypotheses, the robot can be reduced to only four bodies: the traveling plate and the three upper arms. We choose $r = 2/3$ as the optimal

distribution of the three forearm masses, the total virtual mass m_{pt} of the traveling plate based on the hypotheses is calculated as:

$$m_{pt} = m_p + m_L + 3(1-r)m_{fa} \tag{3}$$

At the upper arm level, let $m_a = m_{ua} + m_c + rm_{fa}$ denote the mass of each virtual arm. We can define the position of the center of mass in the arm based on the hypothesis as:

$$L_C = L_{ua}\left(\frac{\frac{1}{2}m_{ua} + m_c + rm_{fa}}{m_a}\right) \tag{4}$$

where L_{ua} is the length of upper arm, m_p is the mass of the traveling plate and connectors at this plate, m_L, m_{fa}, m_c, m_{ua} are the mass of external load, a forearm, and connectors at an elbow and an upper arm, respectively.

The inertia matrix $\boldsymbol{I}_a$ of the arms in joint space is a diagonal matrix, and the inertia of each arm $I_{a1} = I_{a2} = I_{a3} = I_{ai}$, $i = 1,\ 2,\ 3$, can be defined as:

$$\boldsymbol{I}_a = \mathrm{diag}\{I_{a1}, I_{a2}, I_{a3}\} \text{ with } I_{ai} = \left(\frac{1}{3}m_{ua} + m_c + rm_{fa}\right)L_{ua}^2 \tag{5}$$

There are two kinds of forces that act on the traveling plate: the gravity force $\boldsymbol{G}_p \in \mathbb{R}^3$ and the inertia force $\boldsymbol{F}_p \in \mathbb{R}^3$. They are respectively given by:

$$\boldsymbol{G}_p = m_{pt}\begin{bmatrix}0 & 0 & -g\end{bmatrix}^{\mathrm{T}} \tag{6}$$

$$\boldsymbol{F}_p = m_{pt}\ddot{\boldsymbol{X}} \tag{7}$$

where g is the gravitational acceleration, $\ddot{\boldsymbol{X}}$ is the Cartesian acceleration vector of the traveling plate, with $\ddot{\boldsymbol{X}} \in \mathbb{R}^3$.

We can transform the forces in Equations (6) and (7) into torques to each motor by Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{3\times3}$.

$$\boldsymbol{\Gamma}_{Fp} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{F}_p = \boldsymbol{J}^{\mathrm{T}}m_{pt}\ddot{\boldsymbol{X}} \tag{8}$$

$$\boldsymbol{\Gamma}_{Gp} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{G}_p = \boldsymbol{J}^{\mathrm{T}}m_{pt}\begin{bmatrix}0 & 0 & -g\end{bmatrix}^{\mathrm{T}} \tag{9}$$

Let $\varphi_i = \frac{2\pi}{3}(i-1)$ be an angle between $\overrightarrow{OA_i}$ and the y-axis, as shown in Figure 1b, $[x, y, z]^{\mathrm{T}}$ represents the position of the traveling plate in the Cartesian coordinate system. Descriptions of the Jacobian matrix are discussed in more detail in [2] and [3], which is:

$$\boldsymbol{J} = -\begin{bmatrix}s_1^T\\ s_2^T\\ s_3^T\end{bmatrix}^{-1}\begin{bmatrix}s_1^T b_1 & 0 & 0\\ 0 & s_2^T b_2 & 0\\ 0 & 0 & s_3^T b_3\end{bmatrix} \tag{10}$$

where:

$$\begin{gathered} S_i = \begin{bmatrix}x\\ y\\ z\end{bmatrix} - {}_i^R R\left(\begin{bmatrix}0\\ R_b\\ 0\end{bmatrix} + \begin{bmatrix}0\\ L_{ua}\cos(q_i)\\ L_{ua}\sin(q_i)\end{bmatrix}\right) \\ {}_i^R R = \begin{bmatrix}\cos(\varphi_i) & -\sin(\varphi_i) & 0\\ \sin(\varphi_i) & \cos(\varphi_i) & 0\\ 0 & 0 & 1\end{bmatrix}, b_i = {}_i^R R\begin{bmatrix}0\\ L_{ua}\sin(q_i)\\ -L_{ua}\cos(q_i)\end{bmatrix},\ i = 1,2,3. \end{gathered} \tag{11}$$

The relationships between velocities $\dot{\boldsymbol{X}}, \dot{\boldsymbol{q}}$ and accelerations $\ddot{\boldsymbol{X}}, \ddot{\boldsymbol{q}}$ in Cartesian space and joint space are described as:

$$\dot{\boldsymbol{X}} = \boldsymbol{J}\dot{\boldsymbol{q}} \tag{12}$$

$$\ddot{\boldsymbol{X}} = \boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} \tag{13}$$

According to the virtual work principle [8,10], we can calculate the torque contribution of the Delta manipulator applied to the motors from the equilibrium between the contributions of inertia forces and non-inertia forces as:

$$\boldsymbol{\Gamma}_{\Delta} = \boldsymbol{I}_a\ddot{\boldsymbol{q}} + \boldsymbol{J}^{\mathrm{T}} m_{pt}\left(\boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}\right) - \boldsymbol{J}^{\mathrm{T}}\boldsymbol{G}_p - \boldsymbol{\Gamma}_{Ga} \tag{14}$$

where $\boldsymbol{\Gamma}_{\Delta} = \begin{bmatrix}\tau_{\Delta 1} & \tau_{\Delta 2} & \tau_{\Delta 3}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ is the applied joint torque vector of the nominal Delta manipulator, $\boldsymbol{\Gamma}_{Ga} \in \mathbb{R}^3$ as the torque vector produced by the gravitational force vector of the arms, $\dot{\boldsymbol{J}}$ is the derivative of the Jacobian matrix, $\boldsymbol{q} = \begin{bmatrix}q_1 & q_2 & q_3\end{bmatrix}^{\mathrm{T}}$, $\dot{\boldsymbol{q}} = \begin{bmatrix}\dot{q}_1 & \dot{q}_2 & \dot{q}_3\end{bmatrix}^{\mathrm{T}}$, $\ddot{\boldsymbol{q}} = \begin{bmatrix}\ddot{q}_1 & \ddot{q}_2 & \ddot{q}_3\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ are the angular position, velocity, and acceleration vector of the actuated joint, and $\dot{\boldsymbol{X}}$ is the velocity vector of the traveling plate.

In Equation (14), we state the inverse dynamic model produced by the nominal mass matrix $\hat{\boldsymbol{M}}(\boldsymbol{q}) \in \mathbb{R}^{3\times 3}$, the nominal Coriolis and centrifugal matrix $\hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{3\times 3}$, and the gravity contribution $\hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q}) \in \mathbb{R}^3$ as the equation below:

$$\boldsymbol{\Gamma}_{\Delta} = \hat{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q}) \tag{15}$$

where:

$$\hat{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{I}_a + m_{pt}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} = \begin{bmatrix} I_{a1} & 0 & 0 \\ 0 & I_{a2} & 0 \\ 0 & 0 & I_{a3} \end{bmatrix} + m_{pt}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} \tag{16}$$

$$\hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = m_{pt}\boldsymbol{J}^{\mathrm{T}}\dot{\boldsymbol{J}} \tag{17}$$

$$\hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q}) = -\boldsymbol{J}^{T}\boldsymbol{G}_p - \boldsymbol{\Gamma}_{Ga} = -\boldsymbol{J}^{\mathrm{T}} m_{pt}\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} - m_a L_{C\mathcal{G}}\begin{bmatrix} \cos(q_1) \\ \cos(q_2) \\ \cos(q_3) \end{bmatrix} \tag{18}$$

$$\boldsymbol{\Gamma}_{Ga} = m_a L_{C\mathcal{G}}\begin{bmatrix} \cos(q_1) \\ \cos(q_2) \\ \cos(q_3) \end{bmatrix} \tag{19}$$

3. The Proposed Controller

3.1. The Finite-Time Synergetic Controller (FS) for Delta Manipulator

According to the state variable theory, the nominal nonlinear dynamics of the Delta manipulator (15) can be rewritten as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x})\boldsymbol{\Gamma} \tag{20}$$

$$\begin{cases} \dot{\boldsymbol{x}}_1 = \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 = -\hat{\boldsymbol{M}}^{-1}(\boldsymbol{q})\left(\hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q})\right) + \hat{\boldsymbol{M}}^{-1}(\boldsymbol{q})\boldsymbol{\Gamma} \\ \boldsymbol{y} = \boldsymbol{x}_1 \end{cases} \tag{21}$$

where $\boldsymbol{\Gamma} = \begin{bmatrix}\tau_1 & \tau_2 & \tau_3\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ is the control input vector, $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}} & \boldsymbol{x}_2^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ is the state matrix of the robotic system, and $\boldsymbol{x}_1 = \boldsymbol{q}, \boldsymbol{x}_2 = \dot{\boldsymbol{q}}$.

The synergetic control design procedure can be stated as follows:

Step 1: We choose a macro-variable associated with the number of joints of the robot as $\boldsymbol{\Psi} = \begin{bmatrix}\psi_1 & \psi_2 & \psi_3\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$, which is generally a linear combination of the state variables for constructing a surface as follows:

$$\boldsymbol{M}_1 = \left\{\boldsymbol{\varepsilon} : \boldsymbol{\Psi} = \boldsymbol{s}(\boldsymbol{\varepsilon}), \boldsymbol{s}(\boldsymbol{\varepsilon}) \in \mathbb{R}^3\right\} \tag{22}$$

where $\boldsymbol{\varepsilon} = \begin{bmatrix} \boldsymbol{e} & \dot{\boldsymbol{e}} \end{bmatrix}^T$ represents the tracking error states, $\boldsymbol{q}_d \in \mathbb{R}^3$ and $\dot{\boldsymbol{q}}_d \in \mathbb{R}^3$ are the desired angular position vector, $\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$ and $\dot{\boldsymbol{e}} = \dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}$ are the tracking error and its derivative vector. The control law can force the trajectories to operate on the synergetic manifold, we define $\boldsymbol{s}(\boldsymbol{\varepsilon})$ as a linear combination function of the system states as:

$$\boldsymbol{\Psi} = \boldsymbol{s}(\boldsymbol{\varepsilon}) = \dot{\boldsymbol{e}} + \boldsymbol{\Lambda}\boldsymbol{e} \tag{23}$$

We can find the derivative of the synergetic macro-variable as:

$$\dot{\boldsymbol{\Psi}} = \ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\dot{\boldsymbol{e}} + \hat{\boldsymbol{M}}(\boldsymbol{q})^{-1}\left(\hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q})\right) - \hat{\boldsymbol{M}}(\boldsymbol{q})^{-1}\boldsymbol{\Gamma} \tag{24}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \lambda_3\} \in \mathbb{R}^{3\times 3}$ is a positive-definite, diagonal matrix.

Step 2: The proposed controller is designed here such that it can force the states to approach the manifold $\boldsymbol{M}_1$ smoothly at a finite time by using an evolution constraint of the synergetic manifold according to the following equation:

$$\beta\dot{\boldsymbol{\Psi}}^{\frac{p}{r}} + \boldsymbol{\Psi} = \boldsymbol{0} \Rightarrow \dot{\boldsymbol{\Psi}} = \left(-\beta^{-1}\boldsymbol{\Psi}\right)^{\frac{r}{p}} \tag{25}$$

where β is a positive definite number, which affects the quality of convergence, $\boldsymbol{\Psi}^{\frac{p}{r}} = \begin{bmatrix} \psi^{\frac{p_1}{r_1}} & \psi^{\frac{p_2}{r_2}} & \psi^{\frac{p_3}{r_3}} \end{bmatrix}^T$, p_i and r_i are positive odd numbers that satisfy the condition $1 < \frac{p_i}{r_i} < 2$, $i = 1,\ 2,\ 3$. This constraint can drive the macro-variable $\boldsymbol{\Psi}$ and its derivative $\dot{\boldsymbol{\Psi}}$ to zero at a finite time.

Step 3: Substituting $\dot{\boldsymbol{\Psi}}$ into the constraint of Equation (25) into the nonlinear dynamic model (19), we can obtain the resulting finite-time synergetic control law associated with the nominal Delta manipulator as:

$$\boldsymbol{\Gamma} = \hat{\boldsymbol{\Gamma}}_{\mathrm{FS}} = \hat{\boldsymbol{M}}(\boldsymbol{q})\left(\ddot{\boldsymbol{q}}_d + \left(\frac{\boldsymbol{\Psi}}{\beta}\right)^{\frac{r}{p}} + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}\right) + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q}) \tag{26}$$

Lemma 1. *[14,38] Suppose that a continuous, positive-definite function satisfies the following inequality:*

$$\dot{V}(t) \le -\alpha V^{\eta}(t), with\ \ \forall t \ge 0,\ V(t_0) \ge 0 \tag{27}$$

where $\alpha > 0$, $0 < \eta < 1$ are constants. Then, for any given t_0, $V(t)$ satisfies the following inequality:

$$\frac{V^{1-\eta}(t) - V^{1-\eta}(t_0)}{1-\eta} \le -\alpha(t - t_0) \tag{28}$$

where $t_0 \le t \le t_1$ and $V(t) \equiv 0, \forall t \ge t_1$ with t_1 is the reaching time, given by:

$$t_1 = t_0 + \frac{V^{1-\eta}(t_0)}{\alpha(1-\eta)} \tag{29}$$

Theorem 1. *Consider a nonlinear system as (21) with the control law in (26), the tracking errors converge to zero in a finite time with a convergence rate depending on the selected parameter β.*

Proof of Theorem 1. To demonstrate the asymptotic stability of the system, we define a Lyapunov candidate function $V = \frac{1}{2}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}$ and we have its derivative as:

$$\dot{V} = \boldsymbol{\Psi}^{\mathrm{T}}\dot{\boldsymbol{\Psi}} \tag{30}$$

Substituting the constraint in (25) into (30), and assuming $r_1 = r_2 = r_3 = r,\ p_1 = p_2 = p_3 = p$, then:

$$\begin{aligned}\dot{V} &= \boldsymbol{\Psi}^{\mathrm{T}}\left(-\beta^{-1}\boldsymbol{\Psi}\right)^{\frac{r}{p}} = \left(-\beta^{-1}\right)^{\frac{r}{p}}\left(\boldsymbol{\Psi}^{T}\right)^{\frac{p}{p}}\boldsymbol{\Psi}^{\frac{r}{p}} \\ &= -\left(\beta^{-1}\right)^{\frac{r}{p}}\|\boldsymbol{\Psi}\|^{\frac{p+r}{p}} \\ &\leq -\left(\beta^{-1}\right)^{\frac{r}{p}}2^{\frac{p+r}{2p}}\left(\tfrac{1}{2}\|\boldsymbol{\Psi}\|^{2}\right)^{\frac{p+r}{2p}}\end{aligned} \tag{31}$$

$$\dot{V}(t) \leq -\beta_1^{-1}V^{\frac{p+r}{2p}}(t_0) \xrightarrow{\text{if } 1<\frac{p}{r}<2,\ \beta>0} \dot{V} \leq 0 \tag{32}$$

where $\beta_1^{-1} = \left(\beta^{-1}\right)^{\frac{r}{p}}2^{\frac{p+r}{2p}} \geq 0$, if β and $\eta = \frac{p+r}{2p}$ satisfies Lemma 1. The synergetic manifold $\boldsymbol{\Psi}$ can converge to zero in the finite time t_1, given by:

$$t_1 = \frac{V^{(1-\frac{p+r}{2p})}(t_0)}{\beta_1^{-1}\left(1-\frac{p+r}{2p}\right)} \tag{33}$$

3.2. The PD Adaptive Finite-Time Nonsingular Synergetic Controller (PDAFS) Based on Radial Basis Function Neural Networks

Based on the Delta manipulator (21) and motor dynamics (2), the general equation describing the robot system incorporating the external disturbance is presented as:

$$\begin{cases}\dot{\boldsymbol{x}}_{\boldsymbol{1}} = \boldsymbol{x}_{\boldsymbol{2}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{2}} = -\boldsymbol{M}(\boldsymbol{q})^{-1}\left(\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\Gamma}_g(\boldsymbol{q}) + \boldsymbol{\Gamma}_M + \boldsymbol{\Gamma}_D\right) + \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{\Gamma} \\ \boldsymbol{q} = \boldsymbol{x}_1\end{cases} \tag{34}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{3\times3}$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{3\times3}$ are the actual mass matrix and actual centrifugal and Coriolis matrix. $\boldsymbol{\Gamma}_g(\boldsymbol{q})$, $\boldsymbol{\Gamma}_M$, and $\boldsymbol{\Gamma}_D \in \mathbb{R}^3$ are the actual gravitational, actual motor dynamic, and actual disturbance torque, respectively.

From the dynamic model in (34), we can see that if the dynamic motions of the robot satisfy the desired trajectory, the torque control signal must be calculated precisely. In discrete-time control, the signal controls in (2) and (34) are discrete, quantized, and extremely difficult to achieve accurate torque values. Based on three object-model controls for the motor model, the manipulator model, and a generalized disturbance model in the Delta robot, this paper proposes the PDAFS controller as a combination of PD, finite-time synergetic, and RBF neural network controller.

$$\boldsymbol{\Gamma}_{\mathrm{PDAFS}} = \boldsymbol{\Gamma}_{\mathrm{PD}} + \boldsymbol{\Gamma}_{\mathrm{FS}} + \hat{\boldsymbol{\Gamma}}_D \tag{35}$$

Throughout this paper, the controller structure is designed with the following propositions, used directly in the convergence analysis.

Proposition 1. *A proportional derivative control (PD) estimates the torques, matching the nominal dynamics of DC motors. To guarantee the stability of angular motor positions without load, we can choose* $\boldsymbol{K}_P = \boldsymbol{\Lambda}\boldsymbol{K}_D$ *as:*

$$\boldsymbol{\Gamma}_{\mathrm{PD}} = \boldsymbol{\Gamma}_{\mathrm{M}} = \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_D\dot{\boldsymbol{e}} = \boldsymbol{K}_D\boldsymbol{\Psi} \tag{36}$$

where $\boldsymbol{K}_P = \mathrm{diag}\{\mathrm{k_{P1}}, \mathrm{k_{P2}}, \mathrm{k_{P3}}\} \in \mathbb{R}^{3\times3}$, $\boldsymbol{K}_D = \mathrm{diag}\{\mathrm{k_{D1}}, \mathrm{k_{D2}}, \mathrm{k_{D3}}\} \in \mathbb{R}^{3\times3}$, and $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \lambda_3\} \in \mathbb{R}^{3\times3}$ are positive-definite diagonal matrices, respectively. $\hat{\boldsymbol{\Gamma}}_{\mathrm{PD}} \in \mathbb{R}^3$ is the torque vector to control the three motor without torques of load.

Proposition 2. *To overcome the nominal dynamics of the Delta manipulator, the finite time synergetic control scheme is proposed.*

$$\boldsymbol{\Gamma}_{\mathrm{FS}} = \hat{\boldsymbol{M}}(\boldsymbol{q})\left(\ddot{\boldsymbol{q}}_d + \left(\frac{\boldsymbol{\Psi}}{\beta}\right)^{\frac{r}{p}} + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}\right) + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q}) \tag{37}$$

where $\boldsymbol{\Gamma}_{\mathrm{FS}}$, $\hat{\boldsymbol{M}}(\boldsymbol{q})$, $\hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})$, *and* $\hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q})$ *are calculated as in Section 2.2.*

Proposition 3. *The unknown parameters and external disturbance of the practical Delta robot system consist of disturbances in the motors, the Delta manipulator from the hypothesis in Section 2.2, and an external load. They all affect the tracking errors of joint positions and velocities by applying an unknown torque value to each motor. For simplifying disturbance models, a generalized disturbance model* $\hat{\boldsymbol{\Gamma}}_D = \begin{bmatrix}\hat{\tau}_{D1} & \hat{\tau}_{D2} & \hat{\tau}_{D3}\end{bmatrix}^T$ *with* $\hat{\boldsymbol{\Gamma}}_D \in \mathbb{R}^3$ *is proposed to estimate online three RBF neural networks [39,40] as:*

$$\hat{\tau}_{Di} = \hat{\boldsymbol{W}}_i^{*\mathrm{T}} \boldsymbol{h}_i(e_i, \dot{e}_i) \tag{38}$$

The adaptive law is designed as:

$$\dot{\hat{\boldsymbol{W}}}_i = \gamma \boldsymbol{h}_i(e_i, \dot{e}_i)\psi_i \tag{39}$$

From descriptions of RBF networks in [39,40], we choose the tracking error and its derivative value as the input vector in this paper.

$$h_{ij}(e_i, \dot{e}_i) = \Phi(\|\boldsymbol{E}_i - \boldsymbol{c}_j\|, b_i) = \exp\left(-\frac{\|\boldsymbol{E}_i - \boldsymbol{c}_j\|^2}{2b_i^2}\right) \tag{40}$$

where $\boldsymbol{\Phi}(.)$ is the Gaussian function of the j-th neural network of the i-th joint, $\boldsymbol{E}_i = \begin{bmatrix}e_i & \dot{e}_i\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^2$ is the input vector of the i-th joint, $\boldsymbol{c} = \begin{bmatrix}\boldsymbol{c}_q^{\mathrm{T}} & \boldsymbol{c}_{\dot{q}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} c_{q1} & c_{q2} & c_{qj} & \dots & c_{qm} \\ c_{\dot{q}1} & c_{\dot{q}2} & c_{\dot{q}j} & \dots & c_{\dot{q}m} \end{bmatrix} \in \mathbb{R}^{2\times m}$ represents the center point matrix, $\boldsymbol{W_i}, \boldsymbol{W_i^*} \in \mathbb{R}^3$ are the weight vector and updated weight vector of joint i. $j = 1, 2, \dots, m$ is hidden neural net number in the hidden layer. $\boldsymbol{h}_i(e_i, \dot{e}_i) = \begin{bmatrix}h_{i1} & h_{i2} & h_{ij} & \dots & h_{im}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^m$ is the output vector of hidden layer, with $i = 1, 2, 3$ as the number order of joint, the learning rate is denoted by γ. The structure of RBF network is presented in Figure 2.

The optimal weights are found by using the Gradient descent algorithm and the adaptive law (39)-; the weight vector of the i-th joint is:

$$\boldsymbol{W}_i(t) = \boldsymbol{W}_i(t - \Delta t_s) - \dot{\hat{\boldsymbol{W}}}_i(t)\boldsymbol{W}_i^* = \boldsymbol{W}_i - \dot{\hat{\boldsymbol{W}}}_i \tag{41}$$

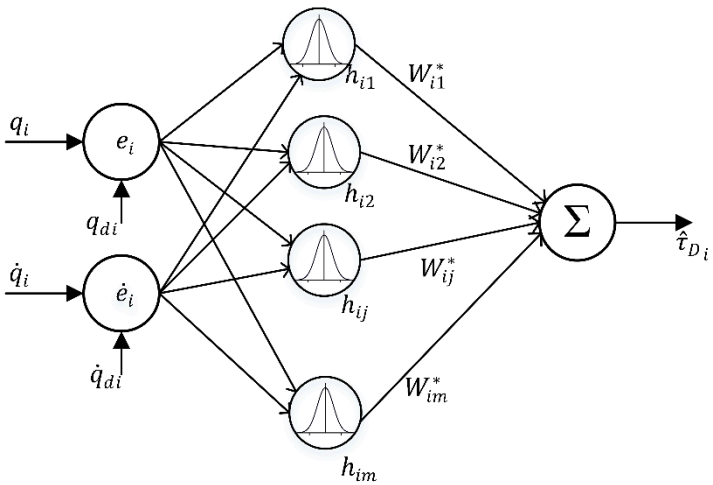


Figure 2. The RFB neural network in this paper.

In total, the proposal controller of the nonlinear Delta robot is proposed as:

$$\begin{aligned}\boldsymbol{\Gamma} &= \widehat{\boldsymbol{M}}(\boldsymbol{q})\left(\ddot{\boldsymbol{q}}_d + \left(\frac{\boldsymbol{\Psi}}{\beta}\right)^{\frac{r}{p}} + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}\right) + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q}) + \boldsymbol{\Gamma}_{\mathrm{PD}} + \hat{\boldsymbol{\Gamma}}_D \\ &= \widehat{\boldsymbol{M}}(\boldsymbol{q})\begin{bmatrix} \ddot{q}_{d1} + \left(\frac{\psi_1}{\beta_1}\right)^{\frac{r_1}{p_1}} + \lambda_1\dot{e}_1 \\ \ddot{q}_{d2} + \left(\frac{\psi_2}{\beta_2}\right)^{\frac{r_2}{p_2}} + \lambda_2\dot{e}_2 \\ \ddot{q}_{d3} + \left(\frac{\psi_3}{\beta_3}\right)^{\frac{r_3}{p_3}} + \lambda_3\dot{e}_3 \end{bmatrix} + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \hat{\Gamma}_g(\boldsymbol{q}) + \begin{bmatrix} k_{d1}\dot{e}_1 + k_{p1}e_1 \\ k_{d2}\dot{e}_2 + k_{p2}e_2 \\ k_{d3}\dot{e}_3 + k_{p3}e_3 \end{bmatrix} + \begin{bmatrix} \boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{h}_1(e_1,\dot{e}_1) \\ \boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{h}_2(e_2,\dot{e}_2) \\ \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{h}_3(e_3,\dot{e}_3) \end{bmatrix} \end{aligned} \tag{42}$$

The system errors and error rates converge to zero in the finite time with the rate convergence depending on the parameters β, p, and r if the control law is selected as Theorem 1 and Section 3.1. The disturbance estimation $\hat{\boldsymbol{\Gamma}}_D$ in (38) is used to approximate the external disturbance and model error compensation in order to minimize the tracking errors. In the following, we prove the stability of the Delta robot model controlled by the PDAFS controller.

Assumption 1. *The uncertainty parameters and external disturbance are unknown but bounded. The characteristics of the rigid robotic manipulator associated with the nominal matrices (16)–(18) are used in the stability analysis and are described in [41,42] as follows:*

1. The inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ is a positive definite symmetric matrix.
2. The matrix $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is a skew symmetric matrix.

Theorem 2. *Assume that the desired trajectories $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d$ are bounded. Let the control input be given by (40) with the estimated torques $\hat{\boldsymbol{\Gamma}}_\Delta$ calculated as (38). Then, we ensure that $\boldsymbol{\Psi}$ is uniformly ultimately bounded.*

Proof: If a Lyapunov candidate function is defined as:

$$V = \frac{1}{2}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{\Psi} + \sum_{i=1}^{3}\frac{1}{2}\widetilde{\boldsymbol{W}}_i^{\mathrm{T}}\gamma^{-1}\widetilde{\boldsymbol{W}}_i \tag{43}$$

Then we can have its derivative as:

$$\begin{aligned}\dot{V} &= \tfrac{1}{2}\boldsymbol{\Psi}^{\mathrm{T}}\dot{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{\Psi} + \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\Psi}} + \sum_{i=1}^{3}\widetilde{\boldsymbol{W}}_i^{\mathrm{T}}\gamma^{-1}\dot{\widetilde{\boldsymbol{W}}}_i \\ &= \tfrac{1}{2}\boldsymbol{\Psi}^{\mathrm{T}}\left[\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right]\boldsymbol{\Psi} + \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\Psi} + \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\Psi}} + \sum_{i=1}^{3}\widetilde{\boldsymbol{W}}_i^{\mathrm{T}}\gamma^{-1}\dot{\widetilde{\boldsymbol{W}}}_i\end{aligned} \tag{44}$$

If we consider structural property 2 in Assumption 1, then one can have $\boldsymbol{\Psi}^{\mathrm{T}}\left[\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right]\boldsymbol{\Psi} = \boldsymbol{0}$ and:

$$\dot{V} = \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\Psi} + \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\Psi}} + \sum_{i=1}^{3}\widetilde{\boldsymbol{W}}_i^{\mathrm{T}}\gamma^{-1}\dot{\widetilde{\boldsymbol{W}}}_i \tag{45}$$

Substituting the manifold surface in (23) and the robot system model in (34) into (24), we have the equation below.

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\Psi}} = \boldsymbol{M}(\boldsymbol{q})\left(\ddot{\boldsymbol{q}}_d + \lambda\dot{\boldsymbol{e}}\right) + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\left(\dot{\boldsymbol{q}}_d + \lambda\boldsymbol{e}\right) + \boldsymbol{\Gamma}_g(\boldsymbol{q}) - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\Psi} + \boldsymbol{\Gamma}_M + \boldsymbol{\Gamma}_D - \boldsymbol{\Gamma} \tag{46}$$

The reference states $\boldsymbol{q}_r$, $\dot{\boldsymbol{q}}_r$ and torque $\mathbf{f}_\mathrm{r}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d)$ to control the generalized Delta robot system are described by Equations (47)–(49):

$$\boldsymbol{q}_r = \dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda e} \tag{47}$$

$$\dot{\boldsymbol{q}}_r = \ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\dot{\boldsymbol{e}} \tag{48}$$

$$\mathbf{f}_\mathrm{r}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{q}_r, \dot{\boldsymbol{q}}_r) = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}_r + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{q}_r + \boldsymbol{\Gamma}_g(\boldsymbol{q}) + \boldsymbol{\Gamma}_M + \boldsymbol{\Gamma}_D \tag{49}$$

Therefore, Formula (46) can be simplified.

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\Psi}} = \mathbf{f}_\mathrm{r} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{\Psi} - \boldsymbol{\Gamma} \tag{50}$$

Substituting (50) into the Lyapunov function derivative in (45), we can get the following formula:

$$\dot{V} = \boldsymbol{\Psi}^\mathrm{T}(\mathbf{f}_\mathrm{r} - \boldsymbol{\Gamma}) + \sum_{i=1}^{3} \widetilde{\boldsymbol{W}}_i^\mathrm{T} \gamma^{-1} \dot{\widetilde{\boldsymbol{W}}}_i \tag{51}$$

We can rewrite the computed torque vector for the Delta robot from the PDAFS controller (42) as:

$$\boldsymbol{\Gamma} = \hat{\mathbf{f}}_\mathrm{r} + \hat{\boldsymbol{M}}(\boldsymbol{q}) \left(\frac{\boldsymbol{\Psi}}{\beta} \right)^{\frac{r}{p}} \tag{52}$$

where:

$$\hat{\mathbf{f}}_\mathrm{r}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{q}_r, \dot{\boldsymbol{q}}_r) = \hat{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}}_r + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{q}_r + \hat{\boldsymbol{\Gamma}}_g(\boldsymbol{q}) - \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{\Psi} + \boldsymbol{\Gamma}_\mathrm{PD} + \hat{\boldsymbol{\Gamma}}_D$$

With the ideal torque to control the Delta robot as $\mathbf{f}_\mathrm{r} = \hat{\mathbf{f}} + \boldsymbol{\Gamma}_D^*$, the computed torque $\hat{\mathbf{f}}_\mathrm{r} = \hat{\mathbf{f}} + \hat{\boldsymbol{\Gamma}}_D$, $\hat{\mathbf{f}}$ is the torque vector matching the nominal models with the computed torque vector for controlling the robot system. For any $\boldsymbol{\Psi}$, we have the difference between $\mathbf{f}_\mathrm{r}$ and $\hat{\mathbf{f}}_\mathrm{r}$ caused by the model errors and external disturbances [13,43], denoted as $\tilde{\boldsymbol{\Gamma}}_D$, which is set as follows:

$$\tilde{\boldsymbol{\Gamma}}_D = \mathbf{f}_\mathrm{r} - \hat{\mathbf{f}}_\mathrm{r} = \boldsymbol{\Gamma}_D^* - \hat{\boldsymbol{\Gamma}}_D \tag{53}$$

Since the disturbances are always bounded with the desired trajectory, assume that the ideal disturbance at joint i is τ_{Di}^* with $i = 1,\ 2,\ 3$, calculated as:

$$\tau_{Di}^* = \boldsymbol{W}_i^{*\mathrm{T}} \boldsymbol{h}_i(e_i, \dot{e}_i) + \varepsilon_{Di} \tag{54}$$

To compare with the estimated values in (38), we state that the error uncertainty parameters $\tilde{\boldsymbol{\Gamma}}_D = \begin{bmatrix} \widetilde{\tau}_{D1} & \widetilde{\tau}_{D2} & \widetilde{\tau}_{D3} \end{bmatrix}^\mathrm{T}$ have an upper bound $\boldsymbol{\varepsilon_D} = \begin{bmatrix} \varepsilon_{D1} & \varepsilon_{D2} & \varepsilon_{D3} \end{bmatrix}^T$, which can be written as:

$$\|\tilde{\boldsymbol{\Gamma}}_D\| = \|\mathbf{f}_\mathrm{r} - \hat{\mathbf{f}}_\mathrm{r}\| = \|\boldsymbol{\Gamma}_D^* - \hat{\boldsymbol{\Gamma}}_D\| < \|\varepsilon_D\| \text{ with } \|\varepsilon_{Di}\| > 0 \tag{55}$$

where $\widetilde{\tau}_{Di} = \tau_{Di}^* - \hat{\tau}_{Di} = \widetilde{\boldsymbol{W}}_i^\mathrm{T} \boldsymbol{h}_i(e_i, \dot{e}_i) + \varepsilon_{Di}$ and $\widetilde{\boldsymbol{W}}_i = \boldsymbol{W}_i^* - \hat{\boldsymbol{W}}_i$.

From (51)–(55), we rewrite Equation (45) as below:

$$\begin{aligned} \dot{V} &= \boldsymbol{\Psi}^\mathrm{T} \left(-\widehat{\boldsymbol{M}}(\boldsymbol{q}) \left(\frac{\boldsymbol{\Psi}}{\beta} \right)^{\frac{r}{p}} \right) + \sum_{i=1}^{3} \widetilde{\boldsymbol{W}}_i^\mathrm{T} \boldsymbol{h}_i(e_i, \dot{e}_i)\psi_i + \sum_{i=1}^{3} \widetilde{\boldsymbol{W}}_i^\mathrm{T} \gamma^{-1} \dot{\widetilde{\boldsymbol{W}}}_i \\ &= \boldsymbol{\Psi}^\mathrm{T} \left(-\widehat{\boldsymbol{M}}(\boldsymbol{q}) \left(\frac{\boldsymbol{\Psi}}{\beta} \right)^{\frac{r}{p}} \right) + \sum_{i=1}^{3} \widetilde{\boldsymbol{W}}_i^\mathrm{T} \left[\boldsymbol{h}_i(e_i, \dot{e}_i)\psi_i - \gamma^{-1} \dot{\widehat{\boldsymbol{W}}}_i \right] \end{aligned} \tag{56}$$

Substituting the adaptive law (37) into (54), yields:

$$\dot{V} = \boldsymbol{\Psi}^\mathrm{T} \left(-\widehat{\boldsymbol{M}}(\boldsymbol{q}) \left(\frac{\boldsymbol{\Psi}}{\beta} \right)^{\frac{r}{p}} \right) \leq 0 \tag{57}$$

According to the Lyapunov theory and the proof in (32), the proposed PDAFS is stable in the finite time for the tracking control of the Delta robot. The equality $\dot{V} = 0$ is satisfied if and only if $\boldsymbol{e} = \dot{\boldsymbol{e}} = 0$. Since the Lyapunov function V is positive definite, and $\dot{V}$ is negative definite, which leads to the conclusion that the proposed controller is globally asymptotically stable from the Lyapunov stability criterion.

Remark 1. *Using the PD controller is realistic, where the PD controller is implemented directly on a microcontroller, and it can stabilize the robot system in the short-term control and guarantee the error boundary of the system and the adaptive finite-time synergetic control (AFS) is calculated on the computer. The torque signals of the AFS can compensate for PD control signals on the microcontroller to ensure the accuracy and stability of the robot in long-term control.*

Remark 2. *There are two main differences between the finite-time synergetic approach in (42) and sliding mode controllers [13,14]. Firstly, the manifold surface of the synergetic controller is constructed by a system errors function and combined with a finite-time constraint function. In contrast, the sliding surface is a function of system errors with an attached finite-time feature from the tracking errors. The other difference is a switching sign function attached to sliding mode control, whereas the asymptotic stability is caused by an exponential convergence function in the proposed controller. In this way, the synergetic approach can reduce the chattering phenomenon and keep the features of robust controllers.*

Remark 3. *In the developed FS in (42), the proposed PDAFS controller can adapt to the existence of external disturbance. The generalized disturbance model based on RBF approximation in the control system can compensate for unknown parameters and significant disturbance influence that assures the error convergence is zero.*

From the remarks, the controller contributions in the PDAFS and their advantages are shown in Table 3.

3.3. Control Design Procedure

The tuning method of the PDAFS can be summarized by following the steps below:

Step 1: Based on the motor model (1) and the parameters in Table 1, the parameters of the PD controller for the motor model can be selected by the PID tuning tool in MATLAB. In a practical experiment, the $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ matrix can be selected by the Zeigler–Nichols tuning method to validate the parameters from the simulation.

Step 2: According to the finite-time synergetic controller, the combined error coefficient matrix $\boldsymbol{\Lambda}$ represents the ratio contribution between angular position error and angular velocity error, the matrix can be set as the ratio of $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$**.** The quality of convergence β can be degraded as small as possible in theory if the system is considered in a continuous domain. However, because of the interval time, the parameter β should be tuned by degradation until the chattering error appears in the practical experiment.

Step 3: Values of angular position error and angular velocity error are evaluated and prepared as the dataset input of the RBF neural networks after implementing the PDFS controller. The synergetic controller can limit the range of the errors, so the remaining errors are associated with unknown disturbances. Based on the boundary of $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$, we can optimize the parameters of RBF NNs for compensated disturbance torques to control the errors into a smaller region without a chattering signal.

Table 3. The contribution and advantages of object-oriented model control in PDAFS.

PDAFS	Torque Contributions	Features	Advantages
PD	DC Motors	Offset-error convergence	Guarantee the stability
FS	Delta manipulator	Finite-time and asymptotic convergence	Manifold surface Chattering noise elimination
RBF networks	Generalized disturbance	Zero-error convergence	Fast function regression Online estimation

3.4. The Velocity Second Order Sliding Mode Observer

The proposed controller defined in (42) requires the knowledge of angular positions and velocities. However, we only use the low-resolution encoders to measure the position and velocity of joints for the experimental Delta robot system (34). Hence, a velocity estimator is essential for the practical controller. Popularly, the velocity values can be calculated by the backward differentiator (BD) technique from the position values.

$$\dot{q}_{i,BD}(t) = \frac{q_i(t) - q_i(t - \Delta t_s)}{\Delta t_s} \tag{58}$$

where $q_i(t)$ and $q(t - \Delta t_s)$ are the present and previous angular positions of the ith joint and Δt_s is the time-delay system.

However, this scheme does not suffer only from the quantization noise due to the low-resolution encoder but also the time-varying delay system in different iterations. To effectively estimate the velocity for this kind of system, a combination of low pass filter (LPF) and second-order exact differentiation via the sliding mode (SOSM) [44,45] is introduced in the experiment results of this paper. The velocity estimator algorithm for the i-th joint is presented as Equations (59)–(61) below.

$$\begin{cases} \dot{q}_{i,SOSM}(t) = v_i(t) \\ v_i(t) = v_{1,i}(t) - \eta_i |\hat{q}_i(t) - q_i(t - \Delta t_s)|^{\frac{1}{2}} \text{sign}(\hat{q}_i(t) - q_i(t - \Delta t_s)) \\ \dot{v}_{1,i}(t) = -\alpha_i \text{sign}(\hat{q}_i(t) - q_i(t - \Delta t_s)) \end{cases} \tag{59}$$

$$\hat{q}_i(t) = \dot{q}_{i,SOSM}(t - \Delta t_s)\Delta t_s + \hat{q}_i(t - \Delta t_s) \tag{60}$$

$$\dot{q}_i(t) = \frac{\dot{q}_{i,SOSM}(t) + \frac{1}{f_c \Delta t_s}\dot{q}_i(t - \Delta t_s)}{\frac{1}{f_c \Delta t_s} + 1} \tag{61}$$

where $\dot{q}_{i,SOSM}$ is the SOSM velocity, $\hat{q}_i$ is the recursive angular position estimation from $\dot{q}_{i,SOSM}$, $\dot{q}_i$ is the estimated velocity, f_c is the cut-off frequency of the low pass filter, and η_i, α_i are the parameters of the SOSM estimator.

Remark 4. *It can be noticed from equation (59) that the velocity estimator using SOSM can achieve finite time error convergence. Boundedness conditions and stable tracking errors with the desired trajectory satisfy the finite-time stability of the sliding mode differentiator [44,46]. Thus, the proposed controller and observer can be designed separately. In the practical assembly system, the angular velocity signal is always quantized by encoder values and contaminated by time-varying delays during estimation, so the control effort leads to chattering. A combination of SOSM and LPF can reduce the chattering phenomenon from the standard (first-order) sliding mode observer [47,48] and preserve the advantage features of the robust exact differentiator.*

4. The Simulation Results

In this section, helix and pick-place characteristics of the nonlinear dynamics of the Delta robot are calculated as the desired trajectory to validate the PDAFS control performance. The planning trajectories followed by a quintic function of time as [6] can

generate smoothing velocities and accelerations, which are applied to reduce vibration in the system. The fifth polynomial function describes the quintic function, as below.

$$\begin{cases} x_H(u) = R\cos(k2\pi u) \\ y_H(u) = R\sin(k2\pi u) \\ z_H(u) = hz_0 u \end{cases} \tag{62}$$

$$\begin{cases} x_P(u) = x_0 + (x_F - x_0)u \\ y_P(u) = y_0 + (y_F - x_0)u \\ z_H(u) = z_0 - 4hu^2 + 4hu \end{cases} \tag{63}$$

$$u(t) = 10\frac{t^3}{t_f^3} - 15\frac{t^4}{t_f^4} + 6\frac{t^5}{t_f^5}, \left(0 \le t \le t_f\right) \tag{64}$$

where x_H, y_H, z_H and x_P, y_P, z_P are the desired positions in the Cartesian coordinate system of the helix and parabolic curves at the determined time. The coordinates x_0, y_0, z_0 and x_F, y_F, z_F are initial and final positions. R is the radius, k is the number circle of the helix, whereas h is the height of parabolic on the z-axis, $u(t)$ is the quintic function of time t, and t_f is the duration of the movement. The characteristic curves of the nominal robot model in Figures 3 and 4 provide the positions in the Cartesian and angular coordinate systems, velocities, accelerations, and desired joint torques for the Delta manipulator. The parameters are calculated via inverse kinematics [2,3] and Equations (12)–(15), respectively.

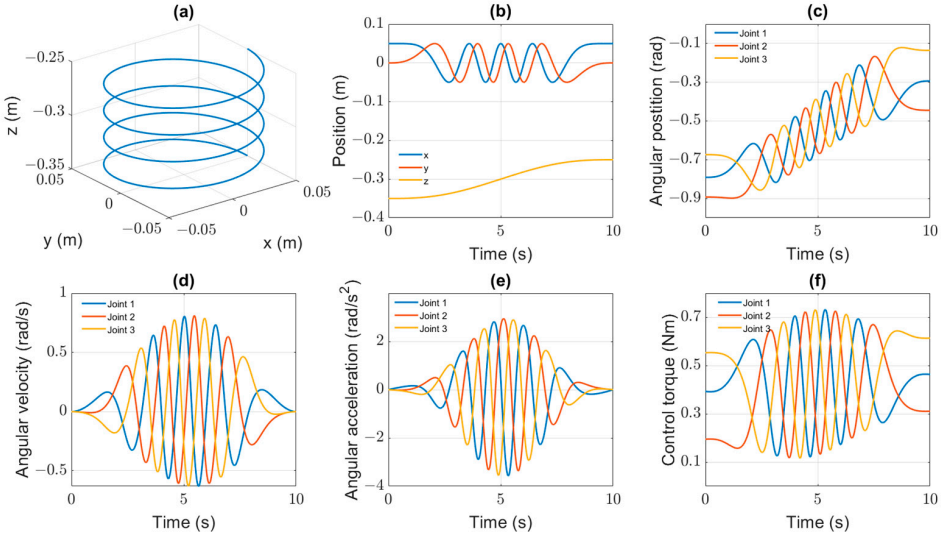


Figure 3. Helix trajectory characteristics: position transitions in workspace (**a**), versus time (**b**); in joint space (**c**), angular velocities (**d**); angular accelerations (**e**); torques of delta manipulator (**f**).

For simulating the dynamic Delta robot model, we use the robot model in Figure 5, which includes the three dynamic DC motors (2) and the Delta manipulator (15). We denote τ_{Li} with $i = 1, 2, 3$ as the load torque applied to the motor in (2). The proposed control scheme of the Delta robot is shown in Figure 6. For demonstrating the effectiveness and advantages of the proposed PDAFS scheme, the simulation results are instructive to compare with that of proportional derivative (PD), PD feedforward (PDFF) [11], PD sliding mode (PDSM) [13], and PDFS regulators. In particular, the performance quantity of the robot trajectory calculated by the i-th joint root mean square error (RMSE) formula, is expressed as:

$$\mathrm{RMSE}_i = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(q_{di_k} - q_{i_k}\right)^2} \tag{65}$$

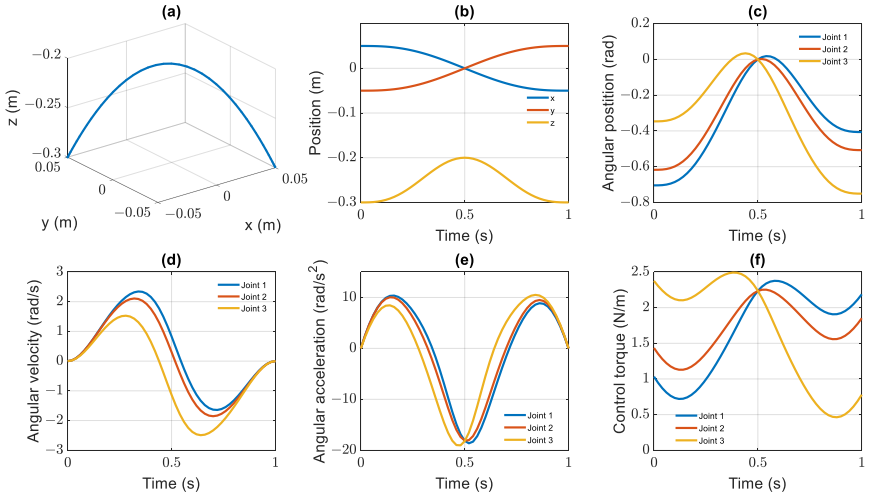


Figure 4. Parabolic characteristics: position transition in workspace (**a**), versus time (**b**); in joint space (**c**), angular velocities (**d**); angular accelerations (**e**); torques of delta manipulator (**f**).

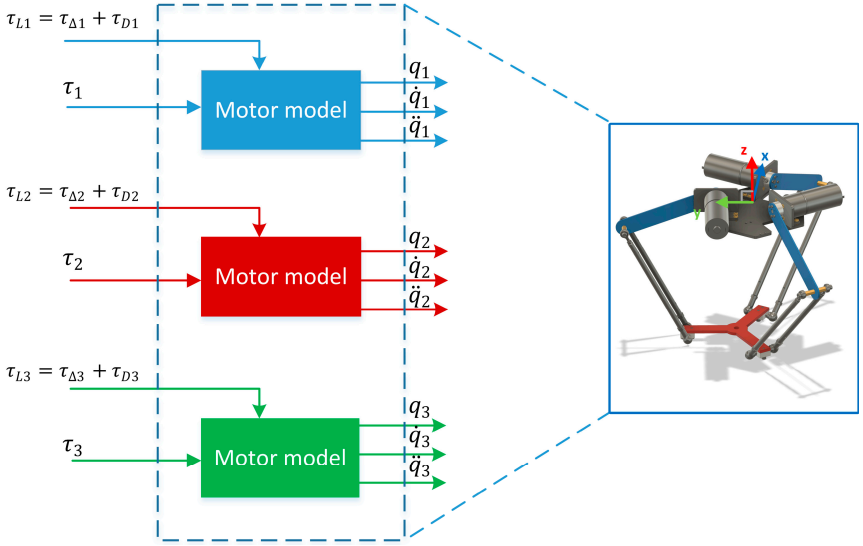


Figure 5. The Delta robot model in this paper.

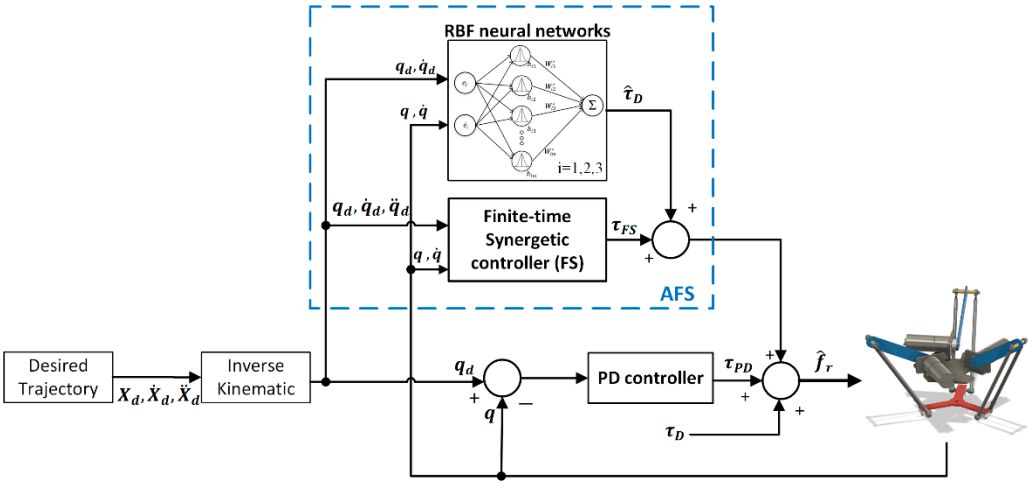


Figure 6. The proposed control scheme.

The controller parameters are chosen constantly in the two cases. Moreover, the adaptability of RBF neural networks is evaluated by the assumption of external disturbance as (66) in the Delta robot model.

$$\begin{cases} \tau_{D1} = 0.4\sin(\dot{q}_1) \\ \tau_{D2} = 0.8\sin(\dot{q}_2) \\ \tau_{D3} = 1.1\sin(\dot{q}_3) \end{cases} \tag{66}$$

With the initial angular error vector such that $\boldsymbol{e}(0) = \begin{bmatrix} e_1(0) & e_2(0) & e_3(0) \end{bmatrix}^T = \begin{bmatrix} -2 & 2 & 5 \end{bmatrix}^T$ in degrees. The desired trajectories for simulation use the information of helix and parabolic trajectory, as shown in Figures 3 and 4, which is generalized by Table 4. The objective of these cases is to validate the effect of the initial condition position and different types of movements on the controllers' ability.

Table 4. The parameters of helix and parabolic trajectory for Delta robot.

Parameter	Helix Trajectory	Parabolic Trajectory
Initial point (x_0, y_0, z_0)	(0.05, 0, −0.35) m	(−0.05, 0.05, −0.3) m
Final point $\left(x_f, y_f, z_f\right)$	(0.05, 0, −0.25) m	(0.05, −0.05, −0.3) m
Number of circles k	4	None
Duration t_f	10 s	1 s
Maximum speed v_{max}	0.24 m/s	0.4 m/s
Maximum acceleration a_{max}	1.2 m/s^2	2.8 m/s^2

The values in Tables 1 and 2 are used as the nominal parameters of the Delta robot. The PDAFS parameters (42) in detail as (36)–(38) are selected as $\boldsymbol{K}_P = \text{diag}\{18, 18, 18\}$ and $\boldsymbol{K}_D = \text{diag}\{1.8, 1.8, 1.8\}$; the constraint in (25) takes the parameters as $\beta^{-1} = 2000$, $r = 5$, $p = 7$. Otherwise, the manifold in (23) takes $\boldsymbol{\Lambda} = \text{diag}\{10, 10, 10\}$. The RBF network has 7 nodes with the center vectors chosen as $\boldsymbol{c}_q = [-0.03, -0.02, -0.01,\ 0,\ 0.01,\ 0.02,\ 0.03]^{\mathrm{T}}$, $\boldsymbol{c}_{\dot{q}} = [-0.3, -0.2, -0.1,\ 0,\ 0.1,\ 0.2,\ 0.3]^{\mathrm{T}}$. The initial neural network weights $\hat{\boldsymbol{W}}_i$ are selected as zeros. The updated weight law is given by (39) and (41) with the adaptation rate $\gamma = 0.0005$. These simulations are implemented with the interval time $\Delta t_s = 10^{-4}$ s. The parameters of the proposed controller and the compared controllers are generalized in Table 5.

Table 5. The parameters of the proposed controller and compared controllers in simulation.

Method	Torque Calculation	Parameters
PD	$\boldsymbol{\Gamma}_{\mathrm{PD}} = \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_D\dot{\boldsymbol{e}}$	$\boldsymbol{K}_P = \text{diag}\{18, 18, 18\}$, $\boldsymbol{K}_D = \text{diag}\{1.8, 1.8, 1.8\}$
PDFF	$\boldsymbol{\Gamma}_{\mathrm{PDFF}} = \boldsymbol{\Gamma}_{\mathrm{PD}} + \boldsymbol{\Gamma}_{\Delta}$	$\boldsymbol{K}_P; \boldsymbol{K}_D$ and Table 2
PDSM	$\boldsymbol{\Gamma}_{\mathrm{PDSM}} = \boldsymbol{\Gamma}_{\mathrm{PD}} + \boldsymbol{\Gamma}_{\Delta} + \boldsymbol{k}\text{sign}(\boldsymbol{\Psi})$	$\boldsymbol{\Lambda} = \text{diag}\{10, 10, 10\}$; $\boldsymbol{k} = \text{diag}\{1.3, 1.3, 1.3\}$
PDFS	$\boldsymbol{\Gamma}_{\mathrm{PDFS}} = \boldsymbol{\Gamma}_{\mathrm{PD}} + \boldsymbol{\Gamma}_{\mathrm{FS}}$	$\boldsymbol{\Lambda} = \text{diag}\{10, 10, 10\}; \beta^{-1} = 2000,\ r = 5,\ p = 7$
PDAFS	$\boldsymbol{\Gamma}_{\mathrm{PDAFS}} = \boldsymbol{\Gamma}_{\mathrm{PD}} + \boldsymbol{\Gamma}_{\mathrm{FS}} + \hat{\boldsymbol{\Gamma}}_D$	$\boldsymbol{\Lambda} = \text{diag}\{10, 10, 10\}; \beta^{-1} = 2000,\ r = 5,\ p = 7$ $\boldsymbol{c}_q = [-0.03, -0.02, -0.01,\ 0,\ 0.01,\ 0.02,\ 0.03]^{\mathrm{T}}$, $\boldsymbol{c}_{\dot{q}} = [-0.3, -0.2, -0.1,\ 0,\ 0.1,\ 0.2,\ 0.3]^{\mathrm{T}}$, $b_i = 0.02$, $\gamma = 0.0005$

For validating the effectiveness of the PDAFS, the tracking errors in the joint space of the five controllers are illustrated and compared in Figure 7a–c for the helix case and Figure 7d–f for the pick and place object assumption. From the tracking error figures, all controllers provided stable trajectory tracking control. Whereas the PD has a large offset error stability, the dynamic model included in PDFF and PDSM provides a minor improvement in the tracking errors. Thanks to the switching function hidden in SM control, the tracking errors of PDSM are better than PDFF. However, PDFS and PDAFS give more

accurate trajectory tracking control with finite time convergence $t_1 \approx 0.1$ s. In particular, the error convergence of PDFS can achieve a small stable value but approximates the mean one as greater than zero. The proposed controller with the disturbance estimator provides remarkable error convergence to zero. The RMSE calculations are processed when the system becomes stable $(t > 0.2$ s$)$ for all controllers. The RMSE of the PDFS and PDAFS controller in Tables 6 and 7 is lower than the values of the others.

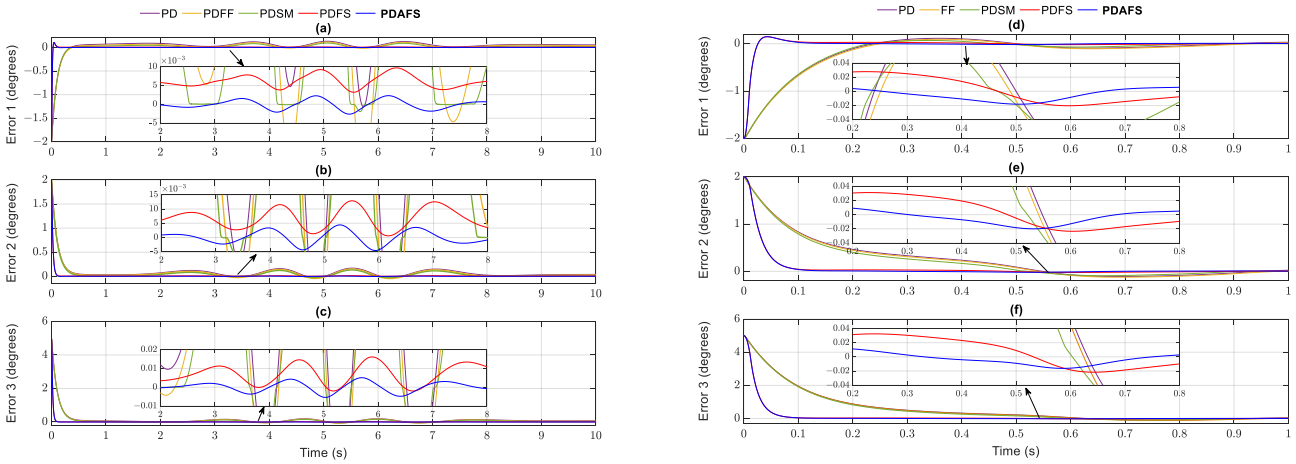


Figure 7. Tracking error comparison among the controllers and PDAFS in the helix and parabolic simulation: (**a**) error of joint 1; (**b**) error of joint 2; (**c**) error of joint 3 in the helix case; (**d**) error of joint 1; (**e**) error of joint 2; (**f**) error of joint 3 in the parabolic case.

Table 6. RMSE Comparison among the controllers and PDAFS in the Helix simulation.

Method	Joint 1 (Degrees)	Joint 2 (Degrees)	Joint 3 (Degrees)
PD	0.0695	0.0800	0.1104
PDFF	0.0559	0.0689	0.0985
PDSM	0.0394	0.0544	0.0841
PDFS	0.0062	0.0066	0.0078
PDAFS	0.0011	0.0018	0.0021

Table 7. RMSE Comparison among the controllers and PDAFS in the Parabolic simulation.

Method	Joint 1 (Degrees)	Joint 2 (Degrees)	Joint 3 (Degrees)
PD	0.0675	0.1912	0.3012
PDFF	0.0686	0.1843	0.2958
PDSM	0.0426	0.1511	0.2644
PDFS	0.0157	0.0182	0.0189
PDAFS	0.0089	0.0093	0.0080

As can be seen, the increasing speed of joints causes the amplitude of the tracking errors to the peaks. This effort is a reason for the error increase caused by the higher maximum speed and acceleration in the parabolic trajectory compared to the helix trajectory. Furthermore, the corresponding maximum errors of the proposed controller are only [0.0181, 0.02, 0.0162] and [0.0044, 0.0046, 0.0054] in degrees. Indeed, the PDAFS can achieve robust motion characteristics for the Delta robot in the simulation conditions.

The control signal of the controllers is presented in Figure 8. In general, the control signals provided by the five controllers have the approximated transitions, whereas the controllers based on the synergetic method demonstrated the better strength signal of the control outline. The control signals of PDSM show the drawback with the chattering noises when they transit the errors on the manifold surface.

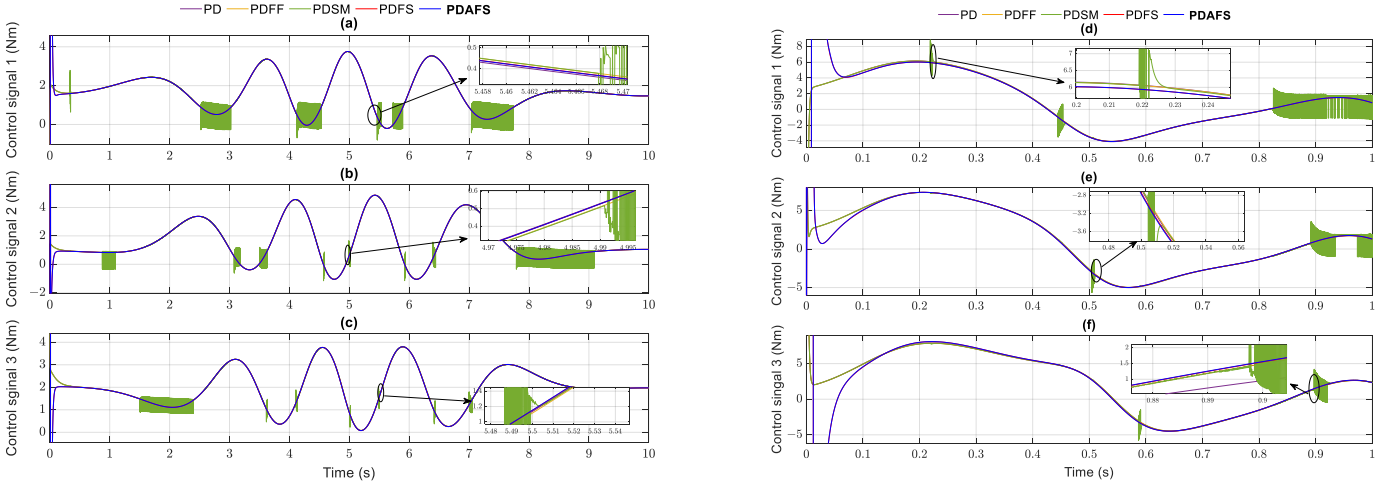


Figure 8. Control signals generated by the controllers in the helix simulation and parabolic simulation: (**a**) control signal of joint 1; (**b**) control signal of joint 2; (**c**) control signal of joint 3 in the helix case; (**d**) error of joint 1; (**e**) error of joint 2; (**f**) error of joint 3 of the parabolic case.

In Figure 9, it is noteworthy that the behavior of external torque estimation is affected by the added external disturbance torques. The effective action of RBF approximation leads the torque compensation to converge to the curves generated by the disturbance assumptions and the remaining torques of the unknown DC motor parameters. The uncertainty disturbance estimation in the control signals leads to the highest performance among the compared controllers in both study cases.

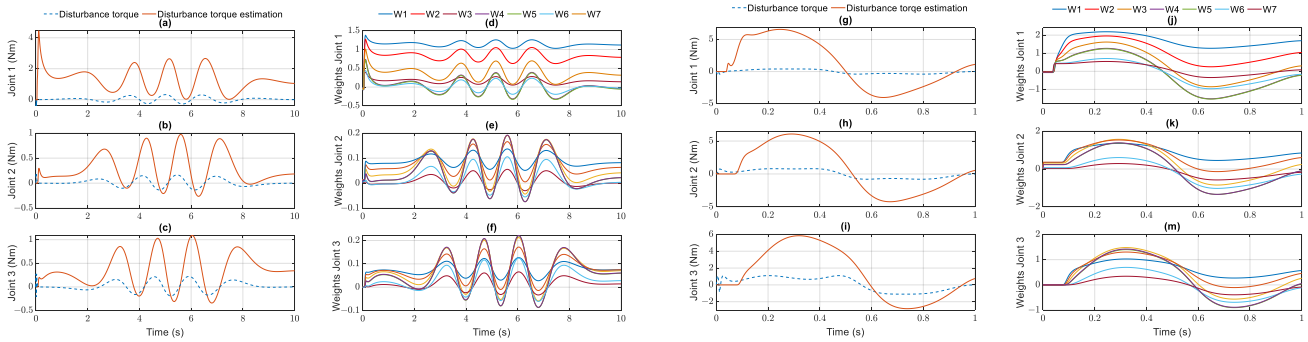


Figure 9. Disturbance torques estimation and weight update of joints from RBF NNs in the helix and in parabolic simulation: disturbance torque and disturbance torque estimation of (**a**) joint 1; (**b**) joint 2; (**c**) joint 3; weights update of (**d**) joint 1; (**e**) joint 2; (**f**) joint 3 in helix case; disturbance torque and disturbance torque estimation of (**g**) joint 1; (**h**) joint 2; (**i**) joint 3; weights update of (**j**) joint 1; (**k**) joint 2; (**m**) joint 3 in parabolic case.

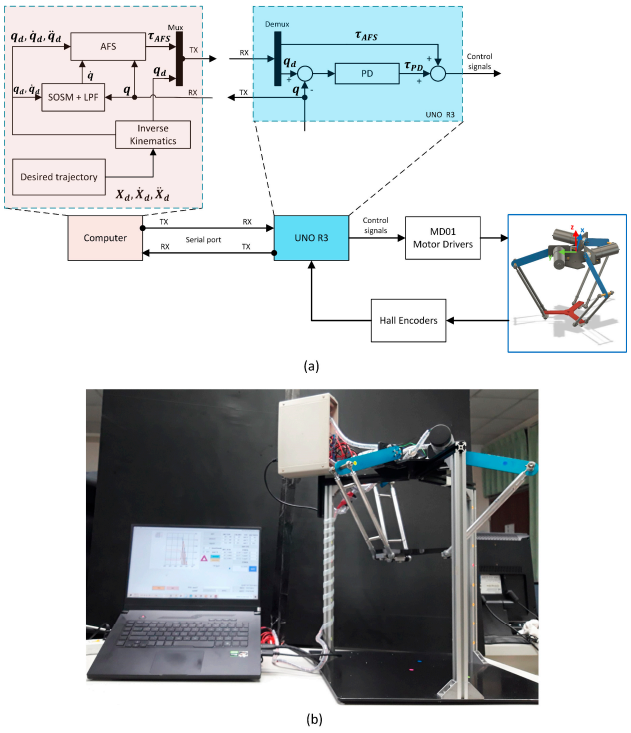
5. Experimental Results

5.1. Sytem Description

In this section, the proposed control scheme is performed and evaluated based on the hardware system in Figure 10a and the experimental setup in Figure 10b.

The robot system with the three motors is driven by the three MD01 high-power motor driver boards. The UNO R3 ATmage328 provides control signals to the motor drivers. For the stability assurance of the Delta robot, the PD controller is programmed directly on the UNO R3 and combines the AFS control signal from a computer to generate the control signal. On the DC motor, the active joint angles are acquired by the mounted three Hall magnetic encoders. The resolution of encoders is 0.367 degrees per pulse based on the specifications. Then, the UNO R3 sends the feedback position signal to the computer for torque calculation in the next iteration. The desired positions of the Delta robot and

compensated torques for the tracking control are verified on the computer. After that, the data signals are transferred via the serial port to the microcontroller board. We programmed the AFS controller and GUI control with C# Visual Studio, and the algorithm runs on a Ryzen 7 4800 HS CPU, 16 GB Ram, and GeForce GTX 1660Ti graphic card. The estimated sampling rate in the real-time system is 0.5 ms on the UNO program. The interval time changes from 14 ms to 20 ms on the computer.



(a)

(b)

Figure 10. The measuring system diagram (**a**) and experiment setup (**b**).

5.2. Velocity Estimation Results

For validating the proposed velocity estimator, the desired helix trajectory function in Equations (62) and (64) is implemented in the system program. Based on the desired trajectory and convergence criterion information in [44,49], we can choose the suitable parameters of the estimator to ensure the error convergence condition and stability of the differentiator. Figure 11 presents the angular velocities estimation using BD and the proposed observer. In particular, the angular velocity estimation obtained by the BD technique shows the large spikes and discontinuous values around the desired velocity. In contrast, the SOSM with LPF estimator does not only give stable fitting values but also reacts with the spike velocities of the BD technique.

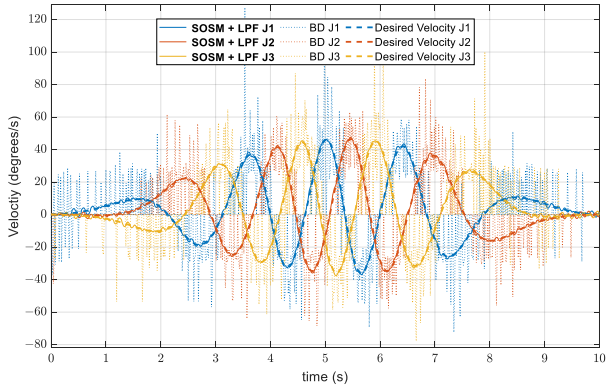


Figure 11. Desired and estimated velocities using BD and SOSM + LPF techniques.

5.3. Case Study Results

From the system description above, it can be seen that the delay time on the computer is much more than the interval time in the simulation. The long time delay combined with the influence of the low-resolution encoders discretizes the control signal values. Thanks to the novel combination in PDAFS, the control signals are generated effectively for improving the Delta robot control performance. In this section, the two case experiments are conducted based on the desired trajectories in (62)–(64). These conditions can reduce unwanted vibration for the Delta robot in the practical environment. In the first case without external load, the performance of the proposed controller is validated based on the features of a robust adaptive controller to an unknown disturbance in the real robot. In the remaining case, an external load is attached to the traveling plate. The purpose of the experiment is to validate the adaptability of the PDAFS with an external disturbance. In our experiments, the PDAFS and the compared controller parameters are turned and presented in Table 8.

Table 8. The parameters of the proposed controller and compared controllers in the experiment.

Method	Torque Calculation	Parameters
PD	$\boldsymbol{\Gamma}_{\mathrm{PD}} = \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_D\dot{\boldsymbol{e}}$	$\boldsymbol{K}_P = \mathrm{diag}\{10, 10, 10\}, \boldsymbol{K}_D = \mathrm{diag}\{0.5, 0.5, 0.5\}$
PDFF	$\boldsymbol{\Gamma}_{\mathrm{PDFF}} = \boldsymbol{\Gamma}_{\mathrm{PD}} + \boldsymbol{\Gamma}_{\Delta}$	$\boldsymbol{K}_P; \boldsymbol{K}_D$ and Table 2
PDSM	$\boldsymbol{\Gamma}_{\mathrm{PDSM}} = \boldsymbol{\Gamma}_{\mathrm{PD}} + \boldsymbol{\Gamma}_{\Delta} + \boldsymbol{k}\mathrm{sign}(\boldsymbol{\Psi})$	$\boldsymbol{\Lambda} = \mathrm{diag}\{20, 20, 20\};\ \boldsymbol{k} = \mathrm{diag}\{3,\ 3,\ 3\}$
PDFS	$\boldsymbol{\Gamma}_{\mathrm{PDFS}} = \boldsymbol{\Gamma}_{\mathrm{PD}} + \boldsymbol{\Gamma}_{\mathrm{FS}}$	$\boldsymbol{\Lambda} = \mathrm{diag}\{20, 20, 20\}; \beta^{-1} = 25,\ r = 5,\ p = 7.$
PDAFS	$\boldsymbol{\Gamma}_{\mathrm{PDAFS}} = \boldsymbol{\Gamma}_{\mathrm{PD}} + \boldsymbol{\Gamma}_{\mathrm{FS}} + \hat{\boldsymbol{\Gamma}}_D$	$\boldsymbol{\Lambda} = \mathrm{diag}\{20, 20, 20\}; \beta^{-1} = 25,\ r = 5,\ p = 7$ $\boldsymbol{c}_q = [-0.1, -0.05, -0.025,\ 0,\ 0.025,\ 0.05,\ 0.1]^{\mathrm{T}},$ $\boldsymbol{c}_{\dot{q}} = [-1, -0.5, -0.25,\ 0,\ 0.25,\ 0.5,\ 1]^{\mathrm{T}},$ $b_i = 0.5, \gamma = 3$

Case 1: Helix and parabolic trajectory without external load

The experiment results of the position and errors in the helix and parabolic path are illustrated in Figures 12 and 13. Figure 14 shows tracking performances in the workspace. The angular errors of PD are stable in the boundedness but non-zero convergent. While the PDFF errors are found with a minor improvement of values, the PDSM errors have some fluctuations in Figures12b,e and 13, but the SM is more stable than PDFF in RMSE comparison. Significantly, the tracking errors of the proposed controllers are under two ticks of the encoder resolution (0.367 degrees/tick) of the RMSE in the helix path and about 1.2 degrees in the second path, the smallest one among those of the other approaches. Moreover, the finite-time convergence property of both synergetic-based controllers is clearly present at the initial times in Figures 12b,d–f and 13d–f. In addition, it can be shown that the effects of the interval time and quadrature encoders in the experiments lead the tracking errors to performance reductions in Tables 9 and 10. The tracking signals from these figures show slight fluctuations compared to those in simulations.

Table 9. RMSE Comparison among the controllers and PDAFS in Helix Case 1.

Method	Joint 1 (Degrees)	Joint 2 (Degrees)	Joint 3 (Degrees)
PD	2.1493	1.5703	2.0829
PDFF	1.9529	1.5272	1.9093
PDSM	1.2824	0.8211	1.3015
PDFS	1.0158	0.8063	1.0859
PDAFS	0.6655	0.5510	0.6872

Table 10. RMSE Comparison among the controllers and PDAFS in Parabolic Case 1.

Method	Joint 1 (Degrees)	Joint 2 (Degrees)	Joint 3 (Degrees)
PD	3.7855	3.5320	3.8794
PDFF	2.8982	3.2263	2.8451
PDSM	2.2124	2.0874	2.7929
PDFS	2.5561	2.4066	2.6639
PDAFS	1.1978	1.1713	1.2120

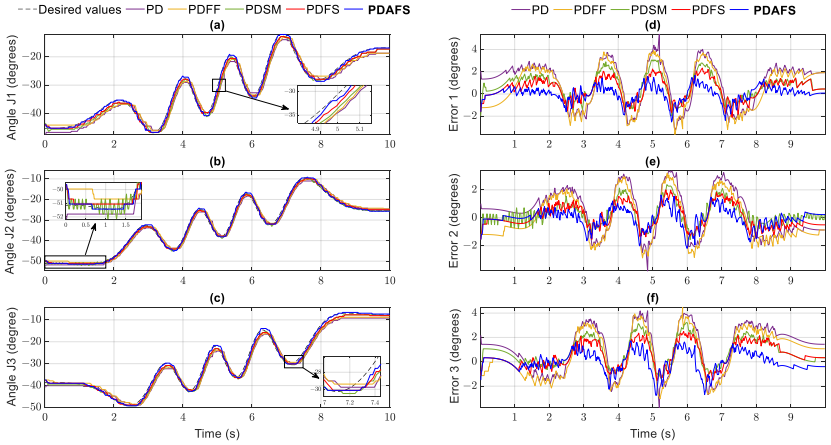


Figure 12. Helix trajectory performance in joint space: (**a**) joint 1; (**b**) joint 2; (**c**) joint 3 and tracking error of joints: (**d**) joint 1; (**e**) joint 2; (**f**) joint 3 in experiment case 1.

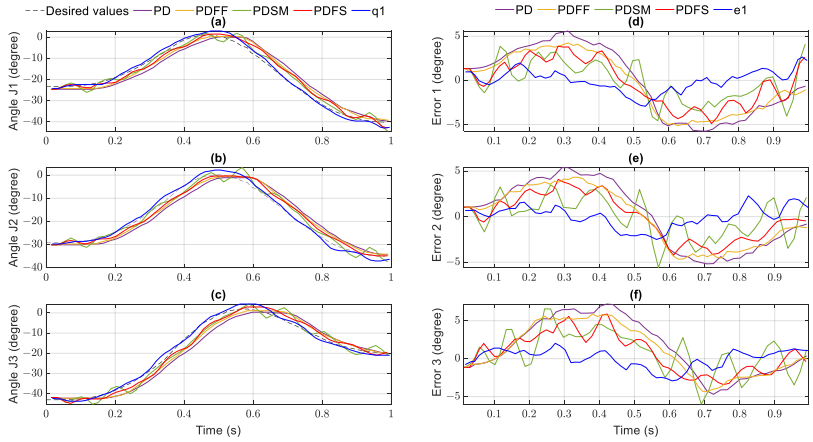
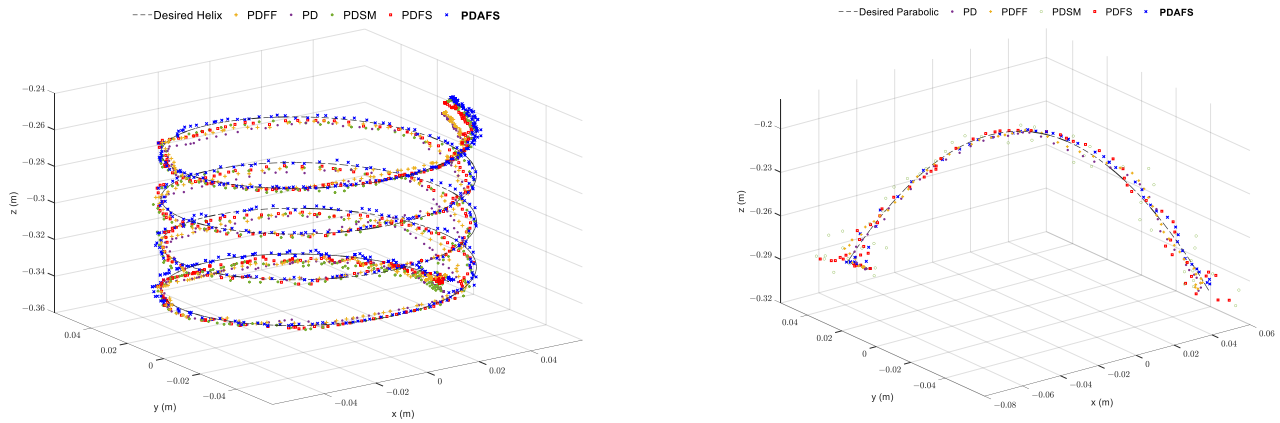


Figure 13. Parabolic trajectory performance in joint space: (**a**) joint 1; (**b**) joint 2; (**c**) joint 3 and tracking error: (**d**) joint 1; (**e**) joint 2; (**f**) joint 3 in the experiment case 1.

It is noteworthy that the behavior of the proposed controller in Figure 15 is very similar to the control torque signals produced by the remaining schemes. The transitions of velocity efforts to the tracking performance and control signals thus consolidating the analyses in the simulation section. Although the PDSM can guarantee stability, some mechanical stresses caused by the on-off signals are witnessed at nearly 1.5 degrees and over 5 degrees, significantly higher than the values of the PDAFS in Figures 12b and 13, respectively. The vibrations are obtained clearly at the SM control signals in Figure 15. Meanwhile, the smoothing of control output from the synergetic algorithm demonstrates the advantages of chattering noise reduction in maintaining the attractiveness of the finite-time manifold constraint.



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

Figure 14. Tracking performance in workspace of case 1: (**a**) Helix trajectory; (**b**) parabolic trajectory.

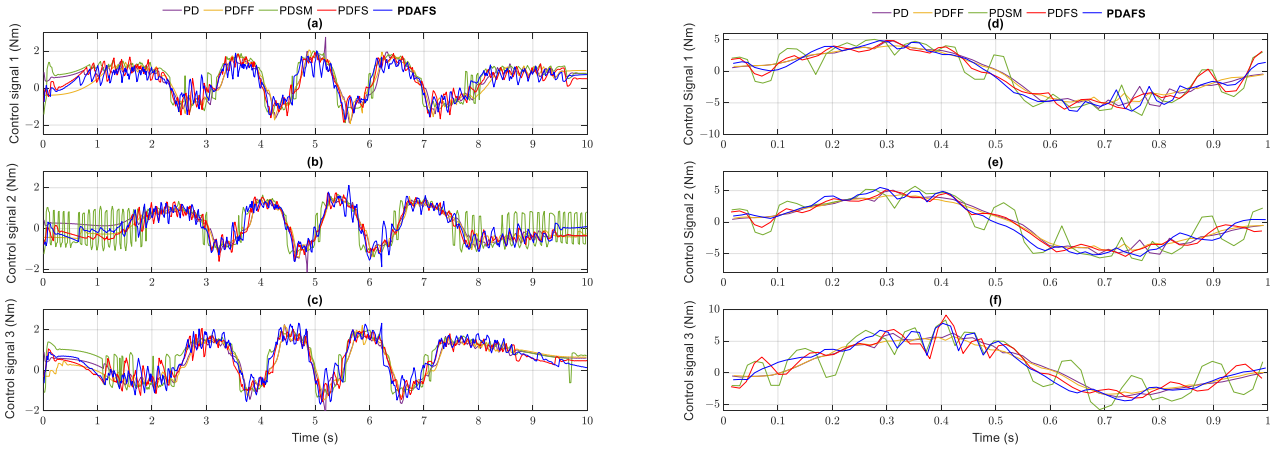


Figure 15. Control signals generated by the controllers in the helix and parabolic of case 1: (**a**) joint 1; (**b**) joint 2; (**c**) joint 3 in the helix trajectory; (**d**) joint 1; (**e**) joint 2; (**f**) joint 3 in the parabolic trajectory.

Although the controller-based dynamic model can provide performance improvement in tracking control, PDFF improves the least values because of the lack of convergent functions in contrast to the robust PDSM or PDFS. The adaptive unknown torque approximation in the proposed approach reduces the effects of the disturbance, resulting in better dynamic performances than the other four methods. The updated weights in Figure 16 reflect the reaction with the dynamic disturbance changes and convergence to the constant values as soon as the system is in the stationary state. The power of torque estimations in the parabolic case witnesses a higher value than that of the helix case, so it is possible to adapt to the fast transitions of external disturbances and ensure the dynamic motions of the Delta parallel the robot.

Case 2: Helix trajectory with the external load

The productiveness of RBF networks in the proposed controller is further performed by an external load attached to the traveling plate of the Delta robot in helix operation. The objective is to evaluate the adaptation and robustness of the proposed approach against model parameter variation. In this case, the traveling plate has added an external object with 0.2 kg. Despite an increasingly small number of errors, the PDAFS algorithm successfully overwhelms the external disturbance to achieve a similar performance compared to others in the no-load case in Figures 17 and 18 and Table 11. The control signal in Figure 18 tends to widen the boundary when the joint velocities increase the speed.

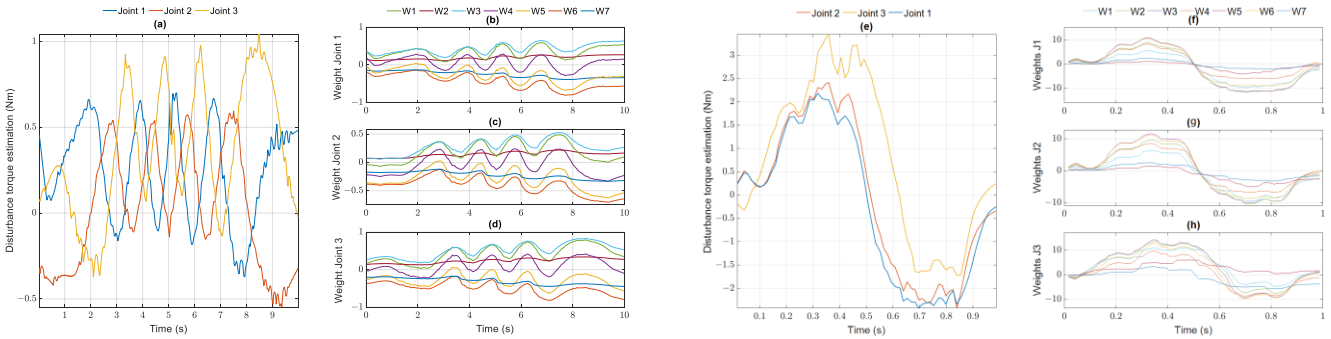


Figure 16. Disturbance torque estimation (**a**) and weight update of (**b**) joint 1; (**c**) joint 2; (**d**) joint 3 from RBFNNs in helix case 1; disturbance torque estimation (**e**) and weight update of (**f**) joint 1; (**g**) joint 2; (**h**) joint 3 from RBFNNs in parabolic case 1.

Table 11. RMSE Comparison among the controllers and PDAFS in Parabolic Case 1.

Method	Joint 1 (Degrees)	Joint 2 (Degrees)	Joint 3 (Degrees)
PDFS (no load)	1.0158	0.8063	1.0859
PDAFS (no load)	0.6655	0.5510	0.6872
PDFS (load)	1.1674	0.8641	1.1363
PDAFS (load)	0.5804	0.6156	0.6103

In addition, Figure 19 demonstrates a similar curve of disturbance torques from the RBF estimators. This action creates the reaction to the new disturbance of the control signal to catch up with the speed of tracking errors and guarantee a robust adaptive system. As a result, the adaptation of the proposed controller is validated based on disturbance torque estimation and compared to no-load cases. Despite the influence of the external disturbances generated by the joint frictions or uncertainty parameters, the PDAFS with RBF NNs attachment has presented more stability than the remaining synergetic control.

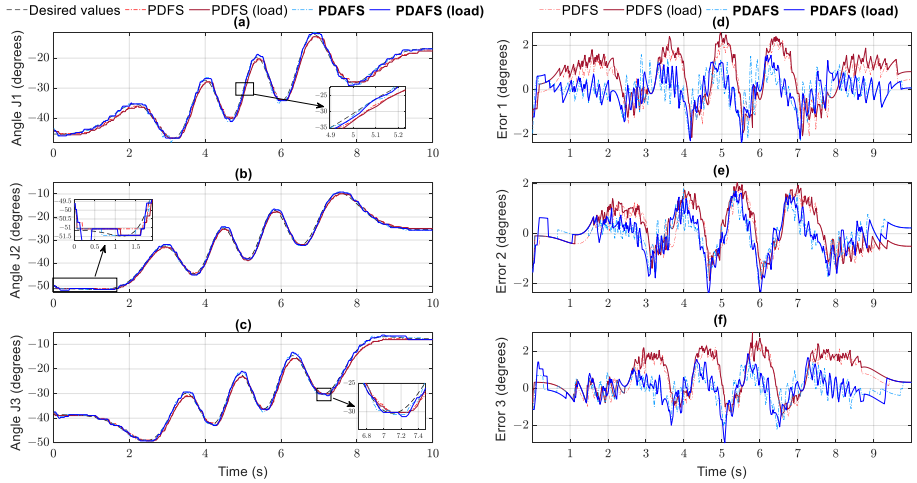


Figure 17. Helix trajectory performance in joint space: (**a**) joint 1; (**b**) joint 2; (**c**) joint 3 and tracking error: (**d**) joint 1; (**e**) joint 2; (**f**) joint 3 in case 2.

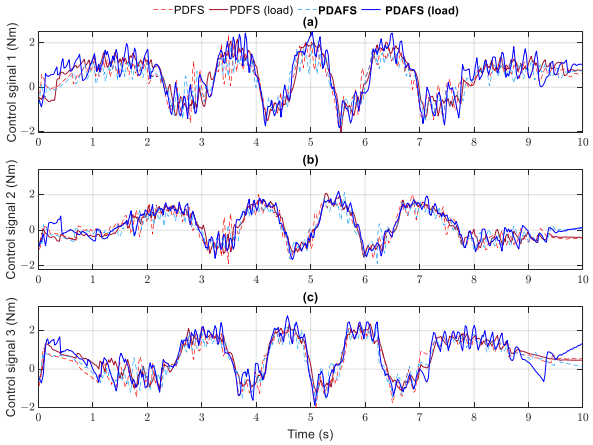


Figure 18. Control signals generated by the PDFS and PDAFS in case 2: joint 1 (**a**), joint 2 (**b**), joint 3 (**c**).

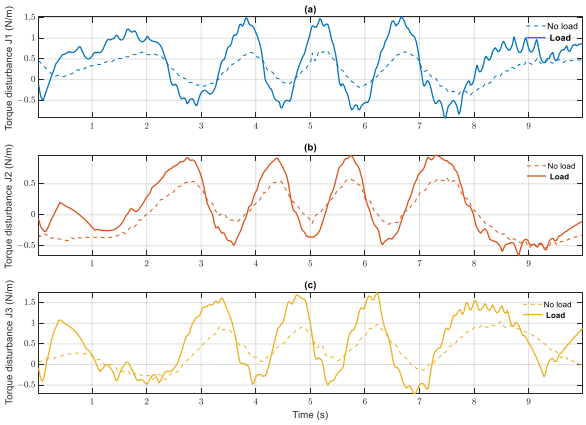


Figure 19. Disturbance torque estimations comparison in case 2: joint 1 (**a**), joint 2 (**b**), joint 3 (**c**).

6. Conclusions

This paper addresses the issue of the robust adaptive control design using the synergetic theorem for the parallel Delta robot. The control structure has been developed by decoupling the objects' control in the system, generated by the DC motors model, robot manipulator model, and generalized uncertain parameters. The stability of the robot system is proven through the Lyapunov stability method. As a result, the proposed controller performs better than the other controllers in the trajectory tracking tasks. The combination structure of the proposed controller can not only achieve the advantages of each algorithm but also overcome the limitations of the compared controller. The differences in the PDAFS control compared with the remaining algorithms in Table 12 shows the effectiveness and adaptive properties of the PDAFS in the presence of disturbance and reduction of the chattering phenomenon. Both simulation and experiment results for the Delta robot confirm all the mentioned benefits of the proposed controller.

Table 12. The property comparison among the controllers and PDAFS.

Method	Features	Ads	Cons
PD	Steady-state error offset	Guarantee the stability Simple to apply	Error offset existence
PDFF [11]	Steady-state error offset with dynamic model	Improve dynamic motion	Exact model parameters requirement
PDSM [13]	Zero-error convergence with dynamic model	Improve dynamic motion	Chattering noise

Table 12. *Cont.*

Method	Features	Ads	Cons
PDFS [19]	Asymptotic convergence with dynamic model	Improve dynamic motion Fast convergence Flexible to apply	Zero-error convergence at infinite time
This work	Zero-error convergence with dynamic model and disturbance consideration	Improve dynamic motion Fast convergence to zero Flexible to apply	Balance contribution of each controller in the system

Author Contributions: Conceptualization, P.-C.P. and Y.-L.K.; methodology, P.-C.P.; software, P.-C.P.; validation, Y.-L.K.; formal analysis, P.-C.P.; investigation, P.-C.P.; resources, Y.-L.K.; data curation, P.-C.P.; writing—original draft preparation, P.-C.P.; writing—review and editing, Y.-L.K.; supervision, Y.-L.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan, grant number MOST 109-2221-E-011-068.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported in part by the Ministry of Science and Technology, Taiwan, under MOST 109-2221-E-011-068.

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

References

1. Taghirad, H.D. *Parallel Robots*, 1st ed.; CRC Press: Cham, Switzerland, 2013; ISBN 9780429097454.
2. Tsai, L.-W. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*; John Wiley & Sons, Inc.: New York, NY, USA, 1999; ISBN 978-0-471-32593-2.
3. López, M.; Castillo, E.; García, G.; Bashir, A. Delta Robot: Inverse, Direct, and Intermediate Jacobians. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2006**, *220*, 103–109. [CrossRef]
4. Gritsenko, I.; Seidakhmet, A.; Abduraimov, A.; Gritsenko, P.; Bekbaganbetov, A. Delta Robot Forward Kinematics Method with One Root. In Proceedings of the 2017 International Conference on Robotics and Automation Sciences (ICRAS), Hong Kong, China, 26–29 August 2017; pp. 39–42.
5. Rosquist, K. *Modelling and Control of a Parallel Kinematic Robot*; Lund University: Lund, Sweden, 2013.
6. Kuo, Y.L.; Huang, P.Y. Experimental and Simulation Studies of Motion Control of a Delta Robot Using a Model-Based Approach. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417738738. [CrossRef]
7. Bortoff, S.A. Object-Oriented Modeling and Control of Delta Robots. In Proceedings of the 2018 IEEE Conference on Control Technology and Applications, CCTA 2018, Copenhagen, Denmark, 21–24 August 2018; pp. 251–258. [CrossRef]
8. Carabin, G.; Scalera, L.; Wongratanaphisan, T.; Vidoni, R. An Energy-Efficient Approach for 3D Printing with a Linear Delta Robot Equipped with Optimal Springs. *Robot. Comput.-Integr. Manuf.* **2021**, *67*, 102045. [CrossRef]
9. Scalera, L.; Carabin, G.; Vidoni, R.; Wongratanaphisan, T. Energy Efficiency in a 4-Dof Parallel Robot Featuring Compliant Elements. *Int. J. Mech. Control* **2019**, *20*, 49–57.
10. Codourey, A. Dynamic Modelling and Mass Matrix Evaluation of the DELTA Parallel Robot for Axes Decoupling Control. *IEEE Int. Conf. Intell. Robot. Syst.* **1996**, *3*, 1211–1218. [CrossRef]
11. Codourey, A. Dynamic Modeling of Parallel Robots for Computed-Torque Control Implementation. *Int. J. Robot. Res.* **1998**, *17*, 1325–1336. [CrossRef]
12. Song, X.; Zhao, Y.; Jin, L.; Zhang, P.; Chen, C. Dynamic Feedforward Control in Decoupling Space for a Four-Degree-of-Freedom Parallel Robot. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881418820451. [CrossRef]
13. Su, T.; Liang, X.; He, G.; Zhao, Q.; Zhao, L. Robust Trajectory Tracking of Delta Parallel Robot Using Sliding Mode Control. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp, 508–512
14. Yu, S.; Yu, X.; Shirinzadeh, B.; Man, Z. Continuous Finite-Time Control for Robotic Manipulators with Terminal Sliding Mode. *Automatica* **2005**, *41*, 1957–1964. [CrossRef]
15. Kolesnikov, A.A. Introduction of Synergetic Control. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014; pp. 3013–3016.

16. Fareh, R.; Khadraoui, S.; Baziyad, M.; Bettayeb, M. Synergetic Workspace Tracking Control for 4-DOF Robot Manipulator. In Proceedings of the 12th International Conference on Computer and Automation Engineering (ICCAE), Sydney, NSW, Australia, 14–16 February 2020; pp. 189–193. [CrossRef]
17. Nusawardhana; Zak, S.H.; Crossley, W.A. Nonlinear Synergetic Optimal Controllers. *J. Guid. Control Dyn.* **2007**, *30*, 1134–1147. [CrossRef]
18. Kuz'menko, A.A. Synchronous Generator Nonlinear Excitation System: Synergetic Sliding Mode Control. In Proceedings of the 2015 International Siberian Conference on Control and Communications (SIBCON), Omsk, Russia, 21–23 May 2015; pp. 1–5.
19. Liu, C.-H.; Hsiao, M.-Y. A Finite Time Synergetic Control Scheme for Robot Manipulators. *Comput. Math. Appl.* **2012**, *64*, 1163–1169. [CrossRef]
20. Zhai, S.; Chen, Q.; Tang, X. Finite-Time Synergetic Control of Mechanical System Based on Model-Free Friction Compensation. In *Lecture Notes in Electrical Engineering*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 337, pp. 275–284, ISBN 9783662464625.
21. Kondratiev, I.; Nikiforov, A.; Veselov, G.; Kolesnikov, A. Synergetic Control for Induction Motor Based Wheel-Drive System. In Proceedings of the 2012 IEEE International Electric Vehicle Conference, Greenville, SC, USA, 4–8 March 2012; pp. 1–7.
22. Kanchanaharuthai, A.; Mujjalinvimut, E. Application of Adaptive Synergetic Control to Power Systems with Superconducting Magnetic Energy Storage System. *Int. J. Innov. Comput. Inf. Control* **2017**, *13*, 1873–1885.
23. Veselov, G.E.; Sclyarov, A.A. Synergetic Approach to Unmanned Air Vehicle Control with "Attractor-Repeller" Strategy of Nondeterministic Obstacles Avoidance. In Proceedings of the CHAOS 2013—6th Chaotic Modeling and Simulation International Conference, Istanbul, Turkey, 11–14 June 2013; pp. 713–726.
24. Boudjedir, C.E.; Boukhetala, D. Adaptive Robust Iterative Learning Control with Application to a Delta Robot. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2021**, *235*, 207–221. [CrossRef]
25. Boudjedir, C.E.; Boukhetala, D.; Bouri, M. Nonlinear PD plus Sliding Mode Control with Application to a Parallel Delta Robot. *J. Electr. Eng.* **2018**, *69*, 329–336. [CrossRef]
26. Boudjedir, C.E.; Bouri, M.; Boukhetala, D. Model-Free Iterative Learning Control with Nonrepetitive Trajectories for Second-Order MIMO Nonlinear Systems—Application to a Delta Robot. *IEEE Trans. Ind. Electron.* **2021**, *68*, 7433–7443. [CrossRef]
27. Shang, W.W.; Cong, S.; Ge, Y. Adaptive Computed Torque Control for a Parallel Manipulator with Redundant Actuation. *Robotica* **2012**, *30*, 457–466. [CrossRef]
28. Zhao, R.; Wu, L.; Chen, Y.H. Robust Control for Nonlinear Delta Parallel Robot with Uncertainty: An Online Estimation Approach. *IEEE Access* **2020**, *8*, 97604–97617. [CrossRef]
29. Castaneda, L.A.; Luviano-Juarez, A.; Chairez, I. Robust Trajectory Tracking of a Delta Robot Through Adaptive Active Disturbance Rejection Control. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 1387–1398. [CrossRef]
30. Linda, O.; Manic, M. Uncertainty-Robust Design of Interval Type-2 Fuzzy Logic Controller for Delta Parallel Robot. *IEEE Trans. Ind. Inform.* **2011**, *7*, 661–670. [CrossRef]
31. Rebai, A.; Guesmi, K.; Hemici, B. Adaptive Fuzzy Synergetic Control for Nonlinear Hysteretic Systems. *Nonlinear Dyn.* **2016**, *86*, 1445–1454. [CrossRef]
32. Escorcia-Hernández, J.M.; Aguilar-Sierra, H.; Aguilar-Mejia, O.; Chemori, A.; Arroyo-Núñez, J.H. A New Adaptive RISE Feedforward Approach Based on Associative Memory Neural Networks for the Control of PKMs. *J. Intell. Robot. Syst.* **2020**, *100*, 827–847. [CrossRef]
33. Huang, S.; Tan, K.K. Intelligent Friction Modeling and Compensation Using Neural Network Approximations. *IEEE Trans. Ind. Electron.* **2012**, *59*, 3342–3349. [CrossRef]
34. Jabbari Asl, H.; Narikiyo, T.; Kawanishi, M. Neural Network-Based Bounded Control of Robotic Exoskeletons without Velocity Measurements. *Control Eng. Pract.* **2018**, *80*, 94–104. [CrossRef]
35. Reg, L. *DC-Motor Modelling and Parameter Identification*; Automation Control, Linkoping University: Linkoping, Sweden, 2017.
36. Urrea Onate, C.; Kern Molina, J. Characterization, Simulation and Implementation of a New Dynamic Model for a DC Servomotor. *IEEE Lat. Am. Trans.* **2014**, *12*, 997–1004. [CrossRef]
37. Turevskiy, A. Estimating DC Motor Parameters. Available online: http://www.mathworks.com/videos/estimating-dc-motor-parameters-97057.html (accessed on 15 March 2020).
38. Moulay, E.; Dambrine, M.; Yeganefar, N.; Perruquetti, W. Finite-Time Stability and Stabilization of Time-Delay Systems. *Syst. Control Lett.* **2008**, *57*, 561–566. [CrossRef]
39. Lee, C.-C.; Chung, P.-C.; Tsai, J.-R.; Chang, C.-I. Robust Radial Basis Function Neural Networks. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **1999**, *29*, 674–685. [CrossRef]
40. Segal, R.; Kothari, M.L.; Madnani, S. Radial Basis Function (RBF) Network Adaptive Power System Stabilizer. *IEEE Trans. Power Syst.* **2000**, *15*, 722–727. [CrossRef]
41. Slotine, J.; Weiping, L. *Applied Nonlinear Control*; Prentice Hall, Pearson: Hoboken, NJ, USA, 1991; ISBN 0130408905.
42. Shang, W.W.; Cong, S.; Li, Z.X.; Jiang, S.L. Augmented Nonlinear PD Controller for a Redundantly Actuated Parallel Manipulator. *Adv. Robot.* **2009**, *23*, 1725–1742. [CrossRef]
43. Safaric, R.; Jezernik, K. Trajectory Tracking Neural Network Controller for a Robot Mechanism and Lyapunov Theory of Stability. In Proceedings of the Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94), Munich, Germany, 12–16 September 1994; Volume 1, pp. 626–633.
44. Levant, A. Robust Exact Differentiation via Sliding Mode Technique. *Automatica* **1998**, *34*, 379–384. [CrossRef]

45. Van, M.; Ge, S.S.; Ren, H. Finite Time Fault Tolerant Control for Robot Manipulators Using Time Delay Estimation and Continuous Nonsingular Fast Terminal Sliding Mode Control. *IEEE Trans. Cybern.* **2017**, *47*, 1681–1693. [CrossRef]
46. Ghanes, M.; Barbot, J.P.; Fridman, L.; Levant, A. A Second Order Sliding Mode Differentiator with a Variable Exponent. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 3300–3305.
47. Slotine, J.J.E.; Hedrick, J.K.; Misawa, E.A. On Sliding Observers for Nonlinear Systems. *J. Dyn. Syst. Meas. Control Trans. ASME* **1987**, *109*, 245–252. [CrossRef]
48. Chan, J.C.L.; Lee, T.H. Sliding Mode Observer-Based Fault-Tolerant Secondary Control of Microgrids. *Electronics* **2020**, *9*, 1417. [CrossRef]
49. Levant, A. Sliding Order and Sliding Accuracy in Sliding Mode Control. *Int. J. Control* **1993**, *58*, 1247–1263. [CrossRef]