Lecture Notes in Networks and Systems 1199

Yo-Ping Huang
Wen-June Wang
Hieu-Giang Le
An-Quoc Hoang *Editors*

Computational Intelligence Methods for Green Technology and Sustainable Development

Proceedings of the International Conference GTSD2024, Volume 2

Springer

Lecture Notes in Networks and Systems 1199

The series "Lecture Notes in Networks and Systems" publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, EI Compendex, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

For proposals from Asia please contact Aninda Bose (aninda.bose@springer.com).

Yo-Ping Huang · Wen-June Wang ·
Hieu-Giang Le · An-Quoc Hoang
Editors

Computational Intelligence Methods for Green Technology and Sustainable Development

Proceedings of the International Conference GTSD2024, Volume 2

Springer

Editors
Yo-Ping Huang
National Penghu University of Science
and Technology
Magong City, Taiwan

Hieu-Giang Le
University of Technology and Education
Ho Chi Minh City, Vietnam

Wen-June Wang
National Central University
Taoyuan, Taiwan

An-Quoc Hoang
Ho Chi Minh City University of Technology
and Education (HCMUTE)
Ho Chi Minh City, Vietnam

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-3-031-76231-4 ISBN 978-3-031-76232-1 (eBook)
https://doi.org/10.1007/978-3-031-76232-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The rapid growth of technology and industrialization has brought great challenges and opportunities for green technology and sustainable development. In this context, the 7th International Conference on Green Technology and Sustainable Development (GTSD 2024) organized on July 25–26, 2024, at Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, Vietnam, emerges as a crucial platform for discussion, innovation, and cooperation.

GTSD 2024 gathers plenty of the leading experts, researchers, and practitioners from many countries including Germany, France, South Korea, Indonesia, Malaysia, Cambodia, Hong Kong, Thailand, India, Sri Lanka, Bangladesh, Poland, Turkey, Vietnam, and so on to express their novel findings and practical solutions for green technology and sustainable development. GTSD 2024 received 312 submissions, and after completing the peer-review process, 220 papers have been selected to be presented at GTSD 2024. The paper documented in this book covers a wide range of topics, from renewable energy systems, smart grid, artificial intelligence, robotics and intelligent systems, and computational intelligence and their applications for sustainable development, climate change mitigation, and environmental policy. The contents of these studies expressed cutting-edge technology and novel ideas related to green technology and provided actionable insights for boosting sustainable development in various sectors.

We hope that the knowledge and innovations documented in this book will be motivations for further research of green technology and sustainable development and related fields.

Finally, we extend our heartfelt thanks to all the participants, authors, reviewers, and organizers who contributed to the success of GTSD 2024. Your contributions are invaluable in our shared journey towards a sustainable world.

Sincerely,

Yo-Ping Huang
Wen-June Wang
Hieu-Giang Le
An-Quoc Hoang

Contents

Design of Adaptive Sliding Mode Controller Based on Neural Network for Robot Manipulator

Le Van Chuong(✉) and Mai The Anh

Vinh University, Vinh City, Nghe An, Vietnam
chuonglv@vinhuni.edu.vn

Abstract. This article presents a method for synthesizing a robot manipulator's adaptive sliding mode controller based on a neural network. In actual working conditions, the robot's dynamic equation has strong nonlinearity, the parameters change uncertainly, and in many cases, the robot is affected by unmeasured external disturbances. Using an RBF neuron network and adaptive control, we propose a solution to approximate and compensate for the uncertain components and external disturbances. The robust control term based on sliding mode control is designed to overcome approximation errors with chattering in the control signal reduced to a minimum. The simulation outcomes indicate that the robot controller suggested in this article possesses high quality, adaptability, and robust resistance to interference.

Keywords: Robot Manipulator · Sliding Mode Control · Adaptive Control · RBF Neural Network

1 Introduction

The surge in demand for heightened product quality and labor efficiency has propelled the widespread integration of robot manipulators in the industry and many other fields. Therefore, improving the ability of robots to operate accurately has become an urgent problem in which control laws play an essential role. Due to the dynamic characteristics with many uncertain factors and often influenced by external disturbances, in recent years, many studies on synthesizing control systems for robots typically combine control methods such as sliding mode control, adaptive control, fuzzy control, and neural networks. Based on existing research, neural networks are widely utilized because they can estimate uncertainty in nonlinear dynamics. Researchers in [1–4] used the RBF neural network to estimate the manipulator uncertainty model to address model uncertainty. Then, these papers propose a neural network-based adaptive terminal sliding mode controller for precise trajectory tracking of the manipulator. Studies [5–7] have shown that fuzzy adaptive controllers can improve robot performance; however, fuzzy controllers depend on expert knowledge, making their application in environments where such knowledge is not available difficult. In the papers [8–10], the authors combined the PD

Y.-P. Huang et al. (Eds.): GTSD 2024, LNNS 1199, pp. 61–68, 2024.
https://doi.org/10.1007/978-3-031-76232-1_6

controller with an adaptive sliding mode controller based on a neural network. To synthesize the sliding mode control law, estimating the upper bound value of the external disturbances components with a constant is necessary. However, in reality, this is only sometimes possible. Besides, if this upper blocking value is significant, the sliding mode control law will cause a strong chattering phenomenon.

In the following sections, our article will present a method for synthesizing a control system for robot manipulators. We design the control law by integrating a traditional PD controller and an adaptive sliding mode controller using the RBF neural network to ensure adaptability and resistance to uncertain factors without knowing the upper blocking value of the external disturbances. Finally, we perform experimental simulations to evaluate the proposed method's effectiveness and draw a conclusion.

2 Problem Formulation

The following second-order nonlinear differential equation can describe the dynamics of a robot manipulator with n- degrees of freedom (n- DOF) [11]:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{f}(\dot{\boldsymbol{q}}) + \boldsymbol{\tau}_{\mathrm{d}}(t) = \boldsymbol{\tau}(t), \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$, $\dot{\boldsymbol{q}} \in \mathbb{R}^n$, $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ are the vectors of joint angular positions, angular velocity, and angular acceleration, respectively; $\boldsymbol{\tau}(t) \in \mathbb{R}^n$ is the vector of control torque produced by the actuators; $\boldsymbol{D}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the positive definite symmetric inertia matrix; $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the matrix that expresses the Coriolis and centrifugal forces; $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the vector of gravitational torques; $\boldsymbol{f}(\dot{\boldsymbol{q}}) \in \mathbb{R}^n$ is the vector of the friction force; $\boldsymbol{\tau}_{\mathrm{d}}(t) \in \mathbb{R}^n$ is the vector generated by external disturbances, which changes slowly, is unmeasurable, and is bounded.

Errors between the robot's trajectory (1) and the desired trajectory $\boldsymbol{q}_{\mathrm{d}}$ exist:

$$\boldsymbol{q}_{\mathrm{e}} = \boldsymbol{q}_{\mathrm{d}} - \boldsymbol{q}. \tag{2}$$

The control goals are $\boldsymbol{q}_{\mathrm{e}} \to 0$ and $\dot{\boldsymbol{q}}_{\mathrm{e}} \to 0$ as $t \to \infty$. The problem is synthesizing a controller for the robot (1) tracking the desired trajectory while resisting uncertainty components in the dynamic model and unmeasured external disturbances.

3 Controller Design

Define the sliding mode function as [12]:

$$\boldsymbol{s} = \dot{\boldsymbol{q}}_{\mathrm{e}} + \boldsymbol{\Lambda}\boldsymbol{q}_{\mathrm{e}}, \tag{3}$$

where $\boldsymbol{s} \in [s_1, s_2, \ldots, s_n]^T$; $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ is the Hurwitz matrix. From (2) and (3) we have:

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{\mathrm{d}} + \boldsymbol{\Lambda}\boldsymbol{q}_{\mathrm{e}} - \boldsymbol{s}. \tag{4}$$

Differentiate both sides of Eq. (4):

$$\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_{\mathrm{d}} + \boldsymbol{\Lambda}\dot{\boldsymbol{q}}_{\mathrm{e}} - \dot{\boldsymbol{s}}. \tag{5}$$

From (1) continuing to transform (5) we have:

$$\boldsymbol{D}^{-1}(\boldsymbol{q})\left[-\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}-\boldsymbol{g}(\boldsymbol{q})-\boldsymbol{f}(\dot{\boldsymbol{q}})-\boldsymbol{\tau}_{\mathrm{d}}(t)+\boldsymbol{\tau}(t)\right]=\ddot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\dot{\boldsymbol{q}}_{\mathrm{e}}-\dot{\boldsymbol{s}}. \tag{6}$$

Continuing with the transformation of (6) and focusing on (4), we obtain the dynamic equation of the system as follows:

$$\boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{s}}=-\boldsymbol{\tau}(t)-\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{s}+\boldsymbol{\delta}(\boldsymbol{x})+\boldsymbol{\tau}_{\mathrm{d}}(t), \tag{7}$$

where $\boldsymbol{\delta}(\boldsymbol{x}) = \boldsymbol{D}(\boldsymbol{q})\left[\ddot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\dot{\boldsymbol{q}}_{\mathrm{e}}\right] + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\left[\dot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\boldsymbol{q}_{\mathrm{e}}\right] + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{f}(\dot{\boldsymbol{q}})$; $\boldsymbol{x} = \left[\boldsymbol{q}_{\mathrm{e}}\ \dot{\boldsymbol{q}}_{\mathrm{e}}\ \boldsymbol{q}_{\mathrm{d}}\ \dot{\boldsymbol{q}}_{\mathrm{d}}\ \ddot{\boldsymbol{q}}_{\mathrm{d}}\right]^{T}$.

The function $\boldsymbol{\delta}(\boldsymbol{x})$ depends on the influence of the robot's mathematical model and the operating environment. In particular, the system may be affected in many cases by unmeasured external disturbances $\boldsymbol{\tau}_{\mathrm{d}}(t)$.

The control law was assumed to include the PD controller, the non-linearity compensating control $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$, the term for compensating unmeasured external disturbances $\hat{\boldsymbol{\tau}}_{\mathrm{d}}(t)$, and the robust control term based on sliding mode control $\boldsymbol{r}(t)$ is proposed as:

$$\boldsymbol{\tau}(t)=\boldsymbol{K}\boldsymbol{s}+\hat{\boldsymbol{\delta}}(\boldsymbol{x})+\hat{\boldsymbol{\tau}}_{\mathrm{d}}(t)-\boldsymbol{r}(t); \tag{8}$$

where $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is a gain matrix, which is a constant matrix satisfying the condition $\boldsymbol{K} = \boldsymbol{K}^{T} > 0$; $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$ is an approximate of $\boldsymbol{\delta}(\boldsymbol{x})$; $\hat{\boldsymbol{\tau}}_{\mathrm{d}}(t)$ is an approximate of $\boldsymbol{\tau}_{\mathrm{d}}(t)$; the control component $\boldsymbol{r}(t) \in \mathbb{R}^{n}$ is designed to overcome approximation errors.

Next, the RBF neural network, which can approximate any non-linear function, is used to approximate the function $\boldsymbol{\delta}(\boldsymbol{x})$. The RBF neural network output is defined as follows [13]:

$$\delta_i(\boldsymbol{x})=\sum_{j=1}^{m}w_{ij}^{*}h_{ij}(\boldsymbol{x})+\varepsilon_i; \tag{9}$$

where $i = 1, 2, \ldots, n; j = 1, 2, \ldots, m$ with m represents the number of basis functions, chosen sufficiently large to ensure approximation error ε_i; w_{ij}^{*} denotes the ideal weight. The basis function is chosen as [13]:

$$h_{ij}(\boldsymbol{x})=\exp\left(\frac{\left\|\boldsymbol{x}-\boldsymbol{c}_{ij}\right\|^{2}}{2b_{ij}^{2}}\right); \tag{10}$$

where $\boldsymbol{c}_{ij}$ is a vector with a dimension equal to the dimension of the vector $\boldsymbol{x}$, representing the center of the ij basis function, and b_{ij} represents the spread of the ij basis function. The approximation of the nonlinear function $\hat{\delta}_i(\boldsymbol{x})$ is established as:

$$\hat{\delta}_i(\boldsymbol{x})=\sum_{j=1}^{m}\hat{w}_{ij}h_{ij}(\boldsymbol{x}); \tag{11}$$

where $\hat{w}_{ij}$ is the estimate of the ideal weighting w_{ij}^{*}. Deviation of the adjusted weight $\hat{w}_{ij}$ compared to the ideal weight w_{ij}^{*} will be:

$$\tilde{w}_{ij}=w_{ij}^{*}-\hat{w}_{ij}. \tag{12}$$

The process of approximating the nonlinear function $\hat{\delta}_i(\boldsymbol{x})$ is adjusting the weight $\hat{w}_{ij}$ of the RBF neural network against the ideal weight w_{ij}^* so that $\tilde{w}_{ij} \to 0$.

From (7) and (8) we have:

$$\boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{s}} = -\boldsymbol{K}\boldsymbol{s} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{s} + \tilde{\boldsymbol{\delta}}(\boldsymbol{x}) + \tilde{\boldsymbol{\tau}}_{\mathrm{d}}(t) + \boldsymbol{r}(t), \tag{13}$$

where

$$\tilde{\boldsymbol{\delta}}(\boldsymbol{x}) = \boldsymbol{\delta}(\boldsymbol{x}) - \hat{\boldsymbol{\delta}}(\boldsymbol{x});\ \tilde{\boldsymbol{\delta}}(\boldsymbol{x}) = \left[\tilde{\delta}_1(\boldsymbol{x}), \tilde{\delta}_2(\boldsymbol{x}), \ldots, \tilde{\delta}_n(\boldsymbol{x})\right]^T; \tag{14}$$

$$\tilde{\boldsymbol{\tau}}_{\mathrm{d}}(t) = \boldsymbol{\tau}_{\mathrm{d}}(t) - \hat{\boldsymbol{\tau}}_{\mathrm{d}}(t);\ \tilde{\boldsymbol{\tau}}_{\mathrm{d}}(t) = [\tilde{\tau}_{\mathrm{d}1}, \tilde{\tau}_{\mathrm{d}2}, \ldots, \tilde{\tau}_{\mathrm{d}n}]^T. \tag{15}$$

From (9), (11) and (12) we have:

$$\tilde{\boldsymbol{\delta}}(\boldsymbol{x}) = \left[\sum_{j=1}^{m} \tilde{w}_{1j} h_{1j}(\boldsymbol{x})\ \sum_{j=1}^{m} \tilde{w}_{2j} h_{2j}(\boldsymbol{x}) \ldots \sum_{j=1}^{m} \tilde{w}_{nj} h_{nj}(\boldsymbol{x})\right]^T + \boldsymbol{\varepsilon}. \tag{16}$$

where $\boldsymbol{\varepsilon} = \left[\varepsilon_1\ \varepsilon_2\ \ldots\ \varepsilon_n\right]^T$; $\|\boldsymbol{\varepsilon}\| \le \varepsilon_{\mathrm{M}}$ and ε_{M} is a very small value; $i = 1, 2, \ldots, n$.

To determine the stability conditions of the system (13), we select a Lyapunov function in the following form:

$$V = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{D}(\boldsymbol{q})\boldsymbol{s} + \frac{1}{2\lambda_1}\sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{w}_{ij}^2 + \frac{1}{2\lambda_2}\sum_{i=1}^{n}\tilde{\tau}_{\mathrm{d}i}^2(t), \tag{17}$$

where λ_1, λ_2 are positive coefficients.

Taking the derivative of both sides of an Eq. (17), we obtain:

$$\dot{V} = \frac{1}{2}\boldsymbol{s}^T\dot{\boldsymbol{D}}(\boldsymbol{q})\boldsymbol{s} + \boldsymbol{s}^T\boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{s}} + \frac{1}{\lambda_1}\sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{w}_{ij}\dot{\tilde{w}}_{ij} + \frac{1}{\lambda_2}\sum_{i=1}^{n}\tilde{\tau}_{\mathrm{d}i}(t)\dot{\tilde{\tau}}_{\mathrm{d}i}(t). \tag{18}$$

By substituting expressions (13), (14), (15), and (16) into (18), we obtain:

$$\begin{aligned}\dot{V} &= \frac{1}{2}\boldsymbol{s}^T\left[\dot{\boldsymbol{D}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right]\boldsymbol{s} - \boldsymbol{s}^T\boldsymbol{K}\boldsymbol{s} + \boldsymbol{s}^T[\boldsymbol{\varepsilon} + \boldsymbol{r}(t)] \\ &\quad + \sum_{i=1}^{n}\sum_{j=1}^{m} s_i\tilde{w}_{ij}h_{ij}(\boldsymbol{x}) + \frac{1}{\lambda_1}\sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{w}_{ij}\dot{\tilde{w}}_{ij} + \sum_{i=1}^{n} s_i\tilde{\tau}_{\mathrm{d}i}(t) + \frac{1}{\lambda_2}\sum_{i=1}^{n}\tilde{\tau}_{\mathrm{d}i}(t)\dot{\tilde{\tau}}_{\mathrm{d}i}(t).\end{aligned} \tag{19}$$

Note that for robot manipulators, we have $\dot{\boldsymbol{D}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ as a skew-symmetric matrix [11], so $\boldsymbol{s}^T\left[\dot{\boldsymbol{D}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right]\boldsymbol{s} = 0$. We select:

$$\dot{\tilde{w}}_{ij} = -\lambda_1 s_i h_{ij}(\boldsymbol{x});\ i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, m; \tag{20}$$

$$\dot{\tilde{\tau}}_{\mathrm{d}i}(t) = -\lambda_2 s_i;\ i = 1, 2, \ldots, n; \tag{21}$$

$$\boldsymbol{r}(t) = -\varepsilon_{\mathrm{M}}\mathrm{sgn}(\boldsymbol{s}). \tag{22}$$

From (20), (21) and (22) continuing to transform expression (19), we have:

$$\dot{V} = -\boldsymbol{s}^T\boldsymbol{K}\boldsymbol{s} + \boldsymbol{s}^T\boldsymbol{\varepsilon} - \boldsymbol{s}^T\varepsilon_{\mathrm{M}}\mathrm{sgn}(\boldsymbol{s}). \tag{23}$$

Because $\boldsymbol{s}^T\boldsymbol{\varepsilon} - \boldsymbol{s}^T\varepsilon_{\mathrm{M}}\mathrm{sgn}(\boldsymbol{s}) = \boldsymbol{s}^T\boldsymbol{\varepsilon} - \varepsilon_{\mathrm{M}}\|\boldsymbol{s}\| \leq 0$, the results final:

$$\dot{V} \leq -\boldsymbol{s}^T\boldsymbol{K}\boldsymbol{s} \leq 0, \tag{24}$$

and system (13) is stable.

From (12), because $w_{ij}^* = \text{const}$ so $\dot{w}_{ij}^* = 0$:

$$\hat{\delta}_i(\boldsymbol{x}) = \sum_{j=1}^{m} \hat{w}_{ij}h_{ij}(\boldsymbol{x});\ \dot{\hat{w}}_{ij} = \lambda_1 s_i h_{ij}(\boldsymbol{x});\ i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, m. \tag{25}$$

From (15), take note that external disturbances change slowly, thus $\dot{\tau}_{\mathrm{d}i} \approx 0$:

$$\dot{\hat{\tau}}_{\mathrm{d}i}(t) = \lambda_2 s_i;\ i = 1, 2, \ldots, n. \tag{26}$$

Thus, with expressions (8), (22), (25), and (26), the article has synthesized a controller for the robot manipulator (1) to track the desired trajectory. With the proposed controller, the system ensures adaptability and robustness with uncertain dynamic components without knowing the upper bounded value of external disturbances in advance. The sliding mode control law (22) only depends on the approximation error of the RBF neural network (arbitrarily small and given), so the chattering phenomenon is reduced to a minimum. The block diagram of the control system for the robot manipulator is shown in Fig. 1. Next, the article performs simulations to evaluate the effectiveness of the proposed robot manipulator control system.

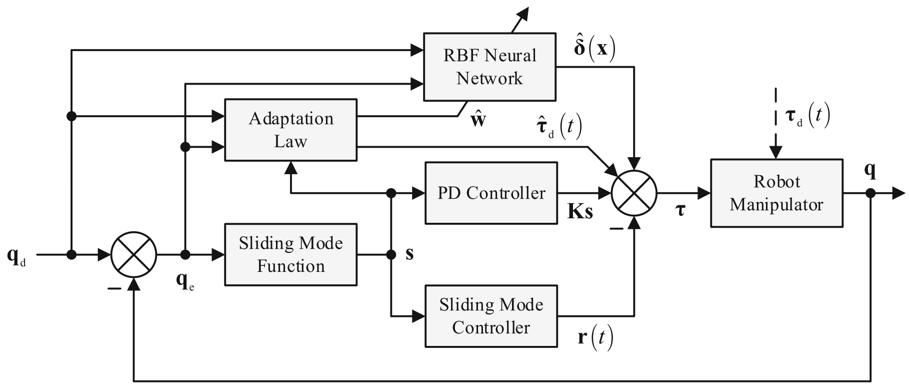

Fig. 1. The control system's block diagram for a robot manipulator.

4 Simulation Example

In this section, the article applies the proposed controller to the 6-DOF Thermo CRS A465 robot manipulator, focusing on the first three degrees during simulation, assuming the fixation of the last three degrees of the robot. By selecting the parameters of the robot manipulator as suggested in [14], the dynamic model is described using Eq. (1).

$$\boldsymbol{D}(\boldsymbol{q}) = \begin{bmatrix} b_3 s_2 s_3 + b_6 c_2^2 + b_7 c_3^2 + b_5 & 0 & 0 \\ 0 & 0.5 b_3 c_{23} + b_{13} & 0.5 b_3 c_{23} + b_{14} \\ 0 & 0.5 b_3 c_{23} + b_{17} & b_{16} \end{bmatrix}; \boldsymbol{g}(\boldsymbol{q}) = \begin{bmatrix} 0 \\ b_8 s_2 + b_9 s_3 \\ b_9 s_3 \end{bmatrix};$$

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} b_1 & b_2 \dot{q}_1 s_2 c_2 + b_3 \dot{q}_1 s_2 c_3 & b_3 \dot{q}_1 s_2 c_3 + b_4 \dot{q}_1 s_3 c_3 \\ 2 b_{11} \dot{q}_1 s_3 c_2 + 2 b_{12} \dot{q}_1 s_2 c_3 - 0.5 b_3 \dot{q}_1 s_{2+3} & 0.5 b_3 \dot{q}_2 s_{3-2} + b_{10} & 0.5 b_3 \dot{q}_3 s_{2-3} \\ 2 b_{12} \dot{q}_1 s_3 c_3 - 0.5 b_3 \dot{q}_1 s_2 c_3 & 0.5 b_3 \dot{q}_2 s_{3-2} - b_{15} & b_{15} \end{bmatrix};$$

with $s_i = \sin(q_i)$; $c_i = \cos(q_i)$; $c_{23} = c_2 c_3 + s_2 s_3$; $s_{2+3} = s_2 c_3 + c_2 s_3$;$s_{3-2} = c_2 s_3 - s_2 c_3$; $s_{2-3} = s_2 c_3 - c_2 s_3$ and the parameters b_k with $k = 1, 2, \ldots, 17$ are given in Table 1.

Table 1. The parameters of the Thermo CRS A465 robot manipulator.

Parameter	Value	Parameter	Value	Parameter	Value
b_1	0.4701	b_7	– 0.0054	b_{13}	0.1991
b_2	0.1094	b_8	– 0.0051	b_{14}	0.0603
b_3	0.0151	b_9	0.0097	b_{15}	0.7218
b_4	0.0591	b_{10}	0.7741	b_{16}	0.1033
b_5	0.0626	b_{11}	0.2345	b_{17}	0.0906
b_6	0.0229	b_{12}	0.0731		

Assume the friction force vector, the external disturbance vector in (1), and the desired trajectory have the form:

$$\boldsymbol{f}(\dot{\boldsymbol{q}}) = \begin{bmatrix} 0.1\mathrm{sgn}(\dot{q}_1) \\ 0.1\mathrm{sgn}(\dot{q}_2) \\ 0.1\mathrm{sgn}(\dot{q}_3) \end{bmatrix}; \boldsymbol{\tau}_\mathrm{d}(t) = \begin{bmatrix} 1.5\sin(0.6t + 0.3) \\ 1.2\cos(0.8t - 0.6) \\ 1.1\sin(0.9t + 0.5) \end{bmatrix}; \boldsymbol{q}_\mathrm{d} = \begin{bmatrix} \cos(0.1t) - 2.1\sin(0.3t) \\ \cos(0.1t) - 1.5\sin(0.5t) \\ 1.8\sin(0.2t) - \cos(0.5t) \end{bmatrix}.$$

By using the control algorithms in (8), (22), (25), and (26), the simulation results are shown in Figs. 2 and 3.

The results in Fig. 2 show that the uncertain components and unmeasured external disturbances have been approximated with an approximation error approaching zero value. Next, Fig. 3 shows that with control law (8), The trajectory of the robot manipulator has adhered to the desired trajectory. Notably, upon approximating and compensating for uncertainty components and external disturbances, the sliding mode control component only depends on the approximation error of the RBF neural network; consequently, the chattering phenomena in control signals have been reduced to a minimum. These simulation outcomes reaffirm our article's proposed control law's accuracy and efficacy.

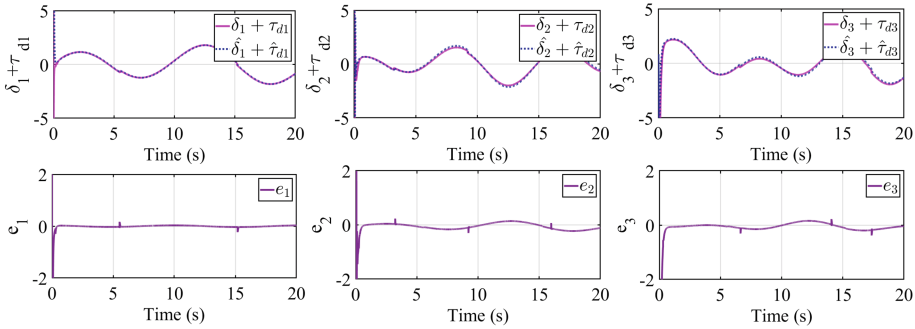

Fig. 2. The result of the approximation and the corresponding error of uncertain components.

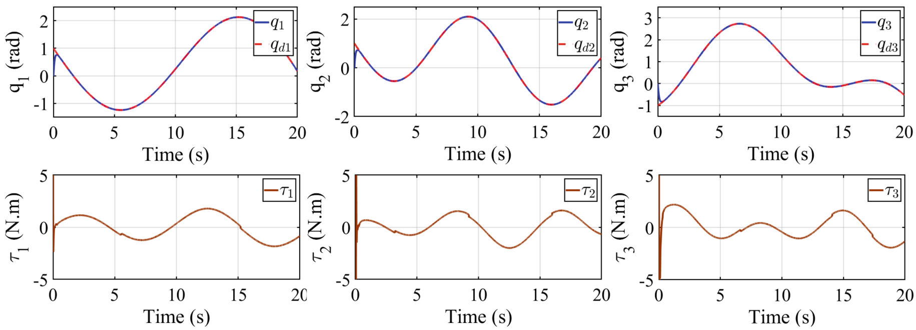

Fig. 3. Trajectory tracking and control signal of the robot manipulator.

5 Conclusion

The article has designed a control system for robot manipulators by integrating traditional PD controllers and adaptive sliding mode controllers using neural networks. We synthesized identification law and compensated for uncertain components and external disturbances using adaptive control theory and RBF neural networks. The advantages of our proposed controller are that it does not need to pre-estimate the limit amplitude of unmeasured external disturbance components. Thanks to this, we achieved robust control laws based on sliding mode control with minimized chattering phenomena. The proposed control system is adaptive, robust, and capable of tracking the desired trajectory with high control quality. Simulation outcomes validate the accuracy and efficacy of the proposed approach.

References

1. Asl, R.M., Hagh, Y.S., Palm, R.: Robust control by adaptive non-singular terminal sliding mode. Eng. Appl. Artif. Intell. **59**, 205–217 (2017)
2. Zhang, S., Yang, P., Kong, L., Chen, W., Fu, Q., Peng, K.: Neural networks-based fault tolerant control of a robot via fast terminal sliding mode. IEEE Transactions on Systems, Man, and Cybernetics: Systems **51**(7), 4091–4101 (2019)

3. Jia, S., Shan, J.: Finite-time trajectory tracking control of space manipulator under actuator saturation. IEEE Trans. Industr. Electron. **67**(3), 2086–2096 (2019)
4. Ren, B., Wang, Y., Chen, J.:Trajectory-tracking-based adaptive neural network sliding mode controller for robot manipulators. J. Comp. Info. Sci. Eng. **20**(3) (2020)
5. Ngo, T., Wang, Y., Mai, T.L., Nguyen, M.H., Chen, J.: Robust adaptive neural-fuzzy network tracking control for robot manipulator. Int. J. Comp. Comm. Contr. **7**(2), 341–352 (2012)
6. Wang, S.: Adaptive fuzzy sliding mode and robust tracking control for manipulators with uncertain dynamics. Complexity 1–9 (2020)
7. Yin, X., Pan, L., Cai, S.: Robust adaptive fuzzy sliding mode trajectory tracking control for serial robotic manipulators. Robotics and Computer-Integrated Manufacturing 72 (2021)
8. Ganie, I., Jagannathan, S.: Adaptive control of robotic manipulators using deep neural networks. IFAC-PapersOnLine. **55**(15), 148–153 (2022)
9. Gierlak, P.: Adaptive position/force control of a robotic manipulator in contact with a flexible and uncertain environment. Robotics **10**(1) (2021)
10. Bao, Y., Thesma, V., Velni, J.M.: Physics-guided and neural network learning-based sliding mode control. IFAC-PapersOnLine **54**(20), 705–710 (2021)
11. Lewis, F.L., Dawson, D.M., Abdallah, C.T.: Robot manipulator control: theory and practice. CRC Press (2003)
12. Utkin, V., Guldner, J., Shi, J.: Sliding mode control in electro-mechanical systems. CRC Press (2017)
13. Huang, S.N., Tan, K.K., Lee, T.H.: A combined PID/adaptive controller for a class of nonlinear systems. Automatica **37**(4), 611–618 (2001)
14. Hoffmann, C., Hashemi, S.M., Abbas, H.S., Werner, H.: Benchmark problem nonlinear control of a 3-DOF robotic manipulator. In: 52nd IEEE Conference on Decision and Control, pp. 5534–5539 (2013)