Trajectory Planning and Control of Cooperative Robotic System for Automated Fiber Placement

Ningyu Zhu

A Thesis

in

The Department

of

Mechanical, Industrial and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Mechanical Engineering) at

Concordia University

Montréal, Québec, Canada

June 2024

© Ningyu Zhu, 2024

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Ningyu Zhu**

Entitled: **Trajectory Planning and Control of Cooperative Robotic System for Automated Fiber Placement**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Yuhong Yan

_____ External Examiner
Dr. Guangjun Liu

_____ External to Program
Dr. (John) Xiupu Zhang

_____ Examiner
Dr. Farjad Shadmehri

_____ Examiner
Dr. Youmin Zhang

_____ Supervisor
Dr. Wen-Fang Xie

Approved by _____
Dr. Muthukumaran Packirisamy, Graduate Program Director

June 25, 2024

_____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Trajectory Planning and Control of Cooperative Robotic System for Automated Fiber Placement

Ningyu Zhu, Ph.D.

Concordia University, 2024

Cooperative robotic system for automated fiber placement (AFP) is a promising solution to fulfill the requirement of manufacturing fiber composites on intricate structures. This project works on the trajectory planning and control of a 13-degree-of-freedom (13-DOF) AFP system. A 1-DOF rotary stage is attached to the end-effector of a 6-Revolute-Spherical-Spherical (6-RSS) parallel robot to hold a Y-shape mandrel, while an AFP head is attached to the end-effector of a 6-DOF serial robot to place fiber with the desired degree. A photogrammetry sensor C-Track 780 can measure the real-time end-effector pose of the robots.

To ensure the desired cooperation performance and limit the communication cost, a distributed control structure with an event-triggered network is developed, based on the measured end-effector pose of the serial robot. An adaptive Kalman filter (AKF) is employed to address uncertain noises in pose estimation. A leader-follower trajectory planning strategy is proposed with the serial robot as the leader and the parallel robot as the follower. A time-jerk optimal trajectory planning scheme is designed for the serial robot considering the kinematic and dynamic constraints. To compensate the serial robot motion and satisfy the AFP geometric constraints, a vision-based trajectory generation approach is developed for the parallel robot.

A position-based visual servoing (PBVS) strategy is proposed for the parallel robot in Cartesian space. To enable the robot to effectively track different trajectories under time-varying conditions, an adaptive sliding mode control method using radial basis function (RBF) neural network is developed to guarantee system robustness and realize the self-tuning of the control gains. In the presence of dynamic uncertainties and external disturbances, a distributed control approach based on adaptive sliding mode controller (ASMC) is developed for the two robots. A deep recurrent neural network (DRNN) is employed to estimate the lumped system uncertainties. The DRNN demonstrates superior learning capability and dynamic adaptability compared to shallow feedforward neural networks. Based on Lyapunov theorem, the stability analyses of the controllers have been done.

The effectiveness and superiority of the proposed algorithms have been validated by simulation and experiment, and comparisons are made with the previous published work.

Acknowledgments

The 4-year Ph.D. study at Concordia University is definitely one of the most memorable experiences in my entire life so far. I want to express my sincere and deepest gratitude to my supervisor, Dr. Wen-Fang Xie. Her guidance, expertise, support and dedication throughout my Ph.D. study have been instrumental in shaping my academic journey and thesis work.

I would like to extend my heartfelt appreciation to my committee members through comprehensive exam, proposal defense and thesis defense. Their insightful feedback and constructive criticism have greatly enriched the quality of my thesis work.

I am also thankful to my colleagues and friends who contributed in various ways to this thesis, whether through discussion, experiment assistance, or encouragement. The friendship and collaboration have made this journey not only intellectually stimulating but also immensely enjoyable.

Most importantly, I am deeply indebted to my family for their unconditional and boundless love, encouragement and understanding during this academic pursuit. I am profoundly grateful for everything they have done for supporting me.

Lastly, I acknowledge the financial support provided by Natural Sciences and Engineering Research Council of Canada (NSERC), without which this research work and the achievements would not have been possible.

To my beloved family

J. P. Zhu, Y. F. Liu and Y. L. Zhu

C. T. Wang, Y. F. Liu and .....

Contents

List of Figures

List of Tables

Chapter 1

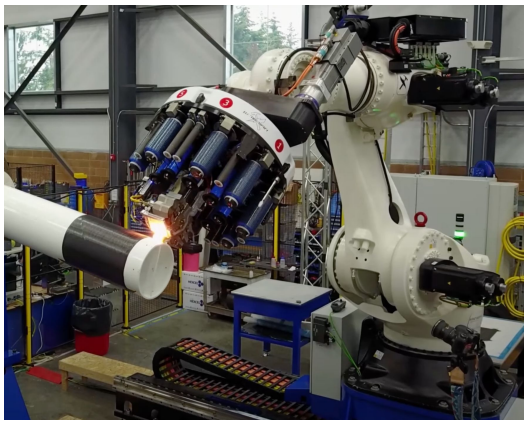
Introduction

1.1 Background

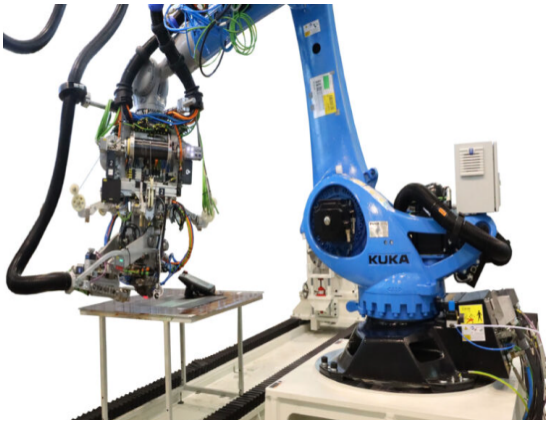
With the characteristics in terms of great impact flexibility, light weight, and high stiffness/strength-to-weight ratios [1], fiber composites have been extensively applied in a number of industrial sectors including aerospace, automobile, renewable energy, and infrastructure [2]. By taking place of conventional materials such as steel, aluminum and titanium alloys, fiber composites contribute a lot to the reduction in operational cost and industrial emissions. According to the 2024 State of the Industry Report from American Composite Manufacturers Association, the demand for lightweight aircraft with high fiber penetration is keeping increasing, i.e. Boeing 787 Dreamliner, Airbus A350XWB and Airbus A380 [3]. Furthermore, transportation can become more energy-efficient using fiber composites in automotive structures. A 10% decrease in the weight of traditional vehicles may result in a 10% boost in fuel economy for electric vehicles and a 6%-8% enhancement for internal combustion engines [4].

Automated fiber placement systems can manufacture fiber composites with less production time, higher accuracy and lower cost, which has become the preferred manufacturing technique for producing composite components in industry. In recent years, it has

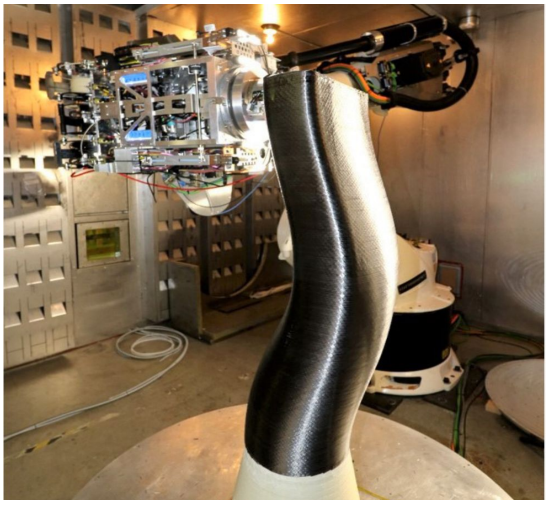
(a)



(b)



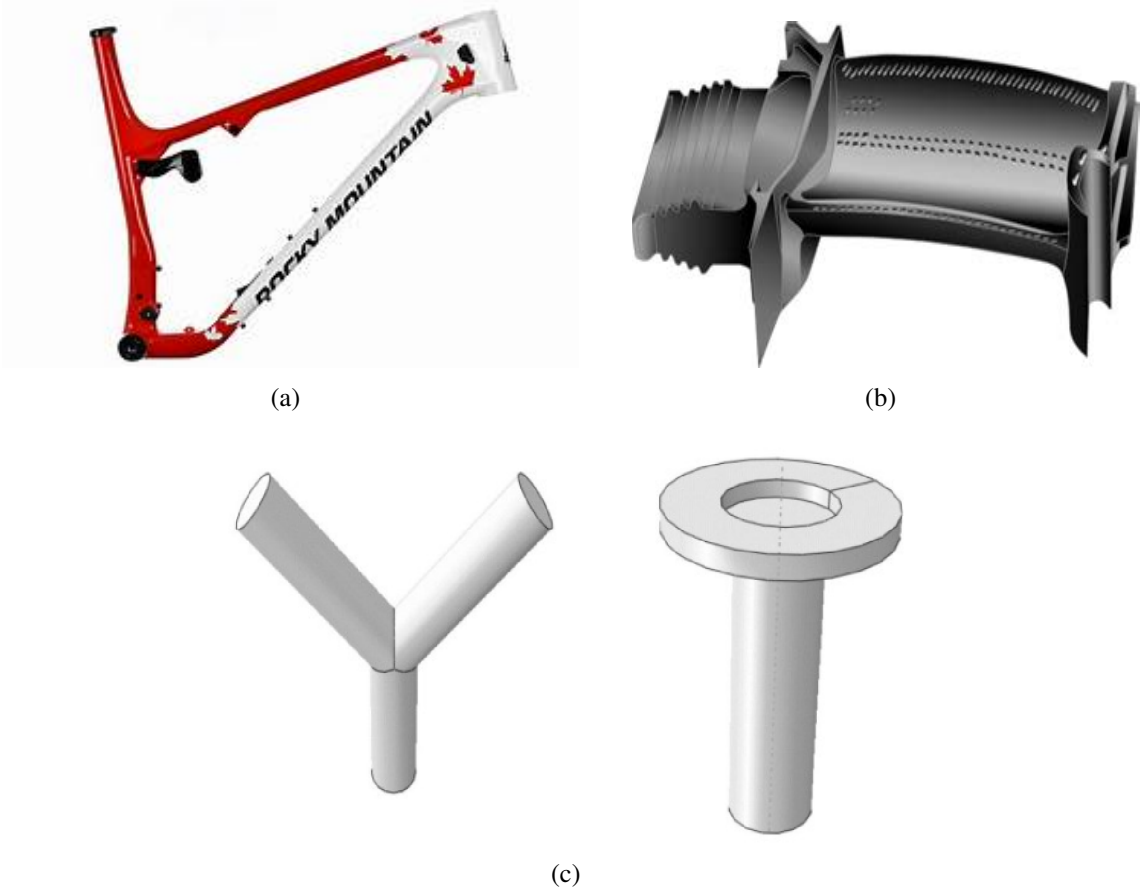
(c)



(d)

Figure 1.1: Automated fiber placement systems in industry. (a) Coriolis C1 [5]. (b) Electroimpact AFP 4.0 machine [6]. (c) Mikrosam Discovery [7]. (d) Scalable Composite Robotic Additive Manufacturing Carbon/Carbon [8].

witnessed the appearance of different kinds of advanced AFP machines, as presented in Figure 1.1. Coriolis C1, complied with aerospace standards, is able to fabricate fibers on the convex or concave panels with double curvatures and complex parts with narrow radius. AFP 4.0 machine manufactured by Electroimpact can realize the modern composite airframe production with fewer lamination tools and lower utility consumption. Mikrosam Discovery employs Kuka Fortec robot to conduct multi-tow or single-tow AFP process with high reliability. Scalable Composite Robotic Additive Manufacturing Carbon/Carbon



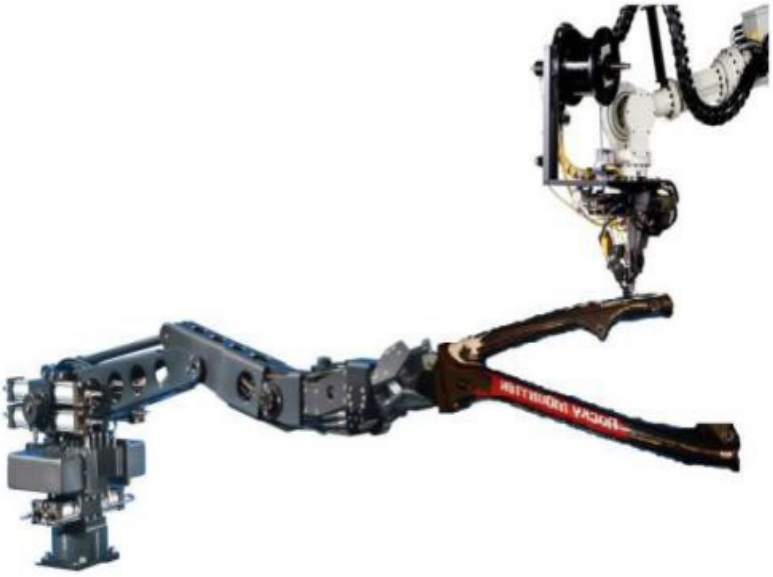
(a) (b)

(c)

Figure 1.2: Closed-loop structures with complex geometry. (a) Bicycle frame. (b) Turbine blade. (c) Y-shape and T-shape Tubes.

from Northrop Grumman is capable of layering continuous fiber-reinforced thermoplastics into components that satisfy the unique demands of hypersonic vehicles.

Nowadays, most of the AFP systems can only produce composite components with open surfaces or simple revolution shapes. Due to the insufficiency of the system DOFs, they are not able to manufacture closed-loop structures with complex geometry, such as bicycle frames, turbine blades of the jet engines, and tubes with T shape or Y shape, as shown in Figure 1.2. To address this issue, some researchers employ an extra serial robot or parallel robot to hold the mandrel to improve the DOFs of the system, as indicated in Figure 1.3. The benefits of large workspace and superior dexterity for fiber layup on complicated geometrical structures can be obtained by integrating two serial robots in the AFP



(a)

(b)

Figure 1.3: AFP systems with two robots [9]. (a) Two serial robots. (b) One serial robot and one parallel robot.

system. However, due to the cantilever structure of the serial robot, the precise positioning ability and load carrying capacity are poor [10]. Parallel robots exhibit better characteristics than serial robots regarding stiffness, load-to-weight ratio, accuracy and speed [11]. By introducing a parallel robot to the AFP machine, better precise positioning capability can be achieved. What's more, parallel robots usually operate in narrower workspace than serial robots, which can reduce the manufacturing cost of the AFP systems.

To enhance the system stiffness and dexterity for dealing with complex closed-loop structures, a 13-DOF cooperative AFP system is developed in [12], as presented in Figure 1.4. A 1-DOF rotary stage is adopted for holding a Y-shape mandrel and attached to the end-effector platform of a 6-RSS parallel robot. An AFP head is manufactured to lay fiber on the mandrel and attached to the end-effector of a 6-DOF serial robot. A photogrammetry sensor C-Track 780 is employed to measure the end-effector pose of the robots. Although cooperative manipulation can achieve good performance in complicated tasks, the control of cooperative multiple robots poses the major challenges to the researchers in composite manufacturing areas. Effective control of industrial robots is a challenging task due to a variety of constraints imposed by the complex kinematics and nonlinear dynamics. Generally, a two-step solution including trajectory planning and trajectory tracking is used to cope with robot control problem. Trajectory planning is to calculate a time history of the robot's desired positions and velocities, while trajectory tracking aims to ensure that the robot's actual positions and velocities match the desired ones. The trajectory planning and tracking control algorithms considering robot kinematics and dynamics should be investigated for the desired AFP tasks.

1.2 Problems and Motivations

With the aim of successfully placing fiber on the Y-shape mandrel, the efficient trajectory planning and accurate tracking control of the cooperative robotic system are essential.

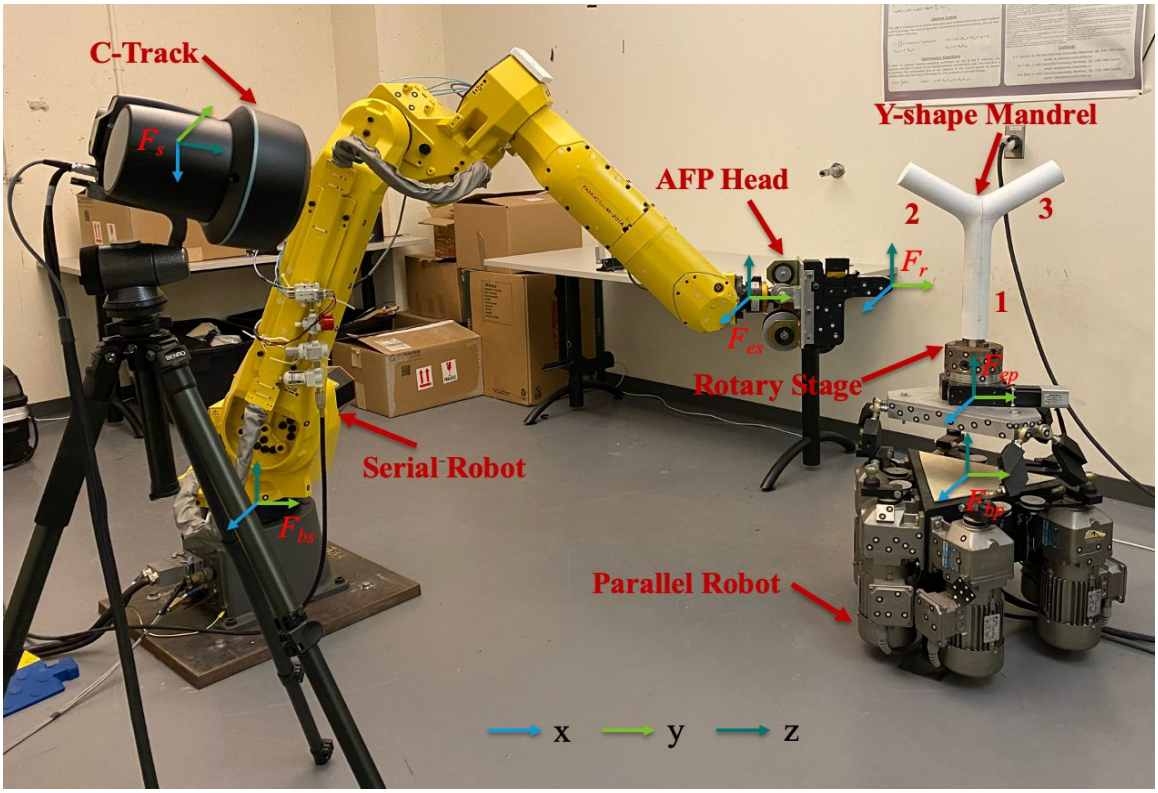


Figure 1.4: 13-DOF cooperative robotic system for AFP.

It is a challenging topic because of the complicated kinematics and nonlinear dynamics of the two different 6-DOF robots and the complexity of the AFP task.

The cooperation tasks performed by multiple manipulators is usually accompanied by the design optimization of a communication network [13]. A functional control structure with real-time data transmission between the 6-DOF serial robot and the 6-RSS parallel robot is crucial for the success of the AFP task. Trajectory planning of the 6-DOF serial robot is significant since the kinematic and dynamic constraints of the robot could cause the interruption of the AFP process [14], which can deteriorate the quality of the composite components. By considering the inherent motion constraints in the trajectory planning, the trajectory of the AFP head may be deviated from the desired fiber path [15]. To ensure that fiber is placed along the desired path, the cooperative motion of the 6-RSS parallel robot is necessary to compensate the serial robot motion for the purpose of satisfying the AFP

geometric constraints.

Due to the fact that the dynamic behavior of high-DOF industrial robots is extremely anisotropic and nonlinear, the kinematics-based controllers are inadequate to satisfy the growing demands regarding accuracy and stability in robotic manufacturing. In this case, it is essential to take into account the dynamics of the robots in the controller development, in order to attain the required performance. With the existence of dynamics modeling errors, external disturbances, and uncertain noises, discrepancies often arise between the real robotic system and its nominal dynamic model [16]. To guarantee good tracking performance, various advanced control strategies have been proposed to address such discrepancies in robotic systems. The mechanical tolerances and deflections in the robot structure can lead to typical differences between a real robot model and its mathematical model [17]. The relatively low accuracy of industrial robots poses a critical obstacle to advanced trajectory planning and control techniques integrating computer-aided simulation and implementation. Therefore, the accuracy enhancement of the robots in the cooperative AFP system is necessary and worth exploring.

1.3 Scope and Objectives

This Ph.D. project focuses on the trajectory planning and control of the 13-DOF cooperative robotic system for fiber placement on the Y-shape mandrel with the desired 0° path. The AFP process generally includes two steps: reference curve generation and offset curve generation, as demonstrated in Figure 1.5. The orientation of the 0° fiber path is parallel to the axis of each cylinder [18]. Once the reference curve is completed, the 1-DOF rotary stage will rotate by a defined small angle, then the offset curve can be generated by offsetting in the orthogonal direction of the reference curve with a constant geodesic distance. In this case, it can avoid the large movements of the serial robot between the placement of two adjacent fiber curves, and the trajectory planning of the two robots can be done with

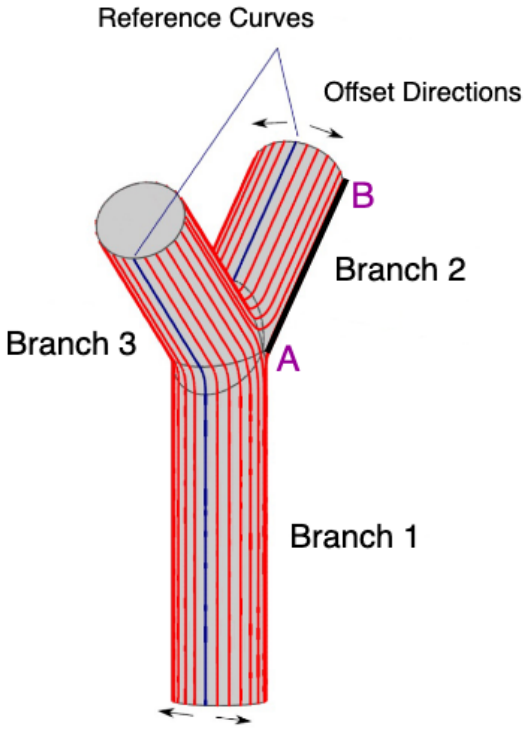


Figure 1.5: 0° fiber path on the Y-shape mandrel.

only little adjustments at each step [19].

In order to fulfill the requirement of the cooperative control for the 13-DOF AFP system, an efficient communication network for motion information exchange between the two robots should be developed. To accomplish the fiber placement process, a cooperative trajectory planning approach needs to be designed to ensure that the motion of the two robots are continuous and smooth without any interruptions, while subjecting to robot inherent constraints and AFP geometric constraints. To obtain desired trajectory tracking performance, advanced nonlinear controllers will be proposed for each robot to cope with system uncertainties. In addition, vision-based pose measurement techniques are adopted for improving the flexibility and accuracy of the robotic system by measuring the end-effector pose of the robots in real time. It plays a vital role in the cooperative planning and control algorithm development in this project.

The research work of this Ph.D. project is carried out in the following phases. First, a distributed control structure with an event-triggered communication network is designed for the cooperative control of the robotic system. Then, a leader-follower trajectory planning scheme with the serial robot as the leader and the parallel robot as the follower is proposed. Additionally, a position-based visual servoing approach based on adaptive sliding mode control is developed for the 6-RSS parallel robot. Furthermore, a distributed adaptive sliding mode control strategy based on deep recurrent network is designed for the two robots. Simulation and experiment are also conducted in each part for the validations of the proposed trajectory planning and control algorithms.

1.4 Contributions

Effective trajectory planning and control schemes for the 13-DOF cooperative robotic system have been developed in this Ph.D. project. The main contributions are summarized as follows.

(1) A distributed control structure with event-triggered condition is proposed for the 6-DOF serial robot and the 6-RSS parallel robot in the AFP system, based on the measured real-time pose of the serial robot. It can guarantee the desired cooperation performance and constrain the workload of the communication network.

(2) With the 6-DOF serial robot as the leader and the 6-RSS parallel robot as the follower, a cooperative trajectory planning strategy is developed. Considering the robot dynamic and kinematic constraints, a time-jerk optimal trajectory planning approach is designed for the serial robot. Relying on the photogrammetry sensor C-Track 780, a vision-based trajectory generation method is proposed for the parallel robot to compensate the serial robot motion to satisfy the AFP geometric constraints.

(3) A position-based visual servoing scheme using adaptive sliding mode control is designed for the 6-RSS parallel robot. An RBF neural network is employed to realize the

auto-tuing of the control gains in the sliding mode controller. The stability of the controller is analyzed utilizing Lyapunov theorem. The trajectory tracking performance of the robot can be enhanced with the real-time pose measurements from the vision sensor and the proposed controller.

(4) A distributed DRNN-based adaptive sliding mode control approach is proposed to address the dynamic uncertainties and external disturbances in the cooperative robotic system. A deep recurrent neural network is designed to approximate the lumped system uncertainties in a model-based sliding mode controller. Based on Lyapunov theorem, the stability analysis of the controller is provided. It demonstrates strong robustness and can realize good trajectory tracking performance for the 6-DOF serial robot and the 6-RSS parallel robot.

(5) Simulation and experiment have been conducted on the 13-DOF cooperative robotic system to verify the effectiveness and superiority of the proposed trajectory planning and tracking control strategies.

1.5 Publications

The following are the publications related to my Ph.D. project and the research work I got involved during my Ph.D. study, including journal papers and conference papers that are published, accepted or submitted.

Journal Paper

• **N. Zhu** and W.-F. Xie, "Distributed adaptive sliding mode control with deep recurrent neural network for cooperative robotic system in automated fiber placement," *IEEE Transactions on Systems, Man and Cybernetics: Systems*. (Submitted)

• R. Zhang, W. Xie, Y. Wang, H. Tan, **N. Zhu**, and L. Song, "Adaptive finite-time coordination control of a multi-robotic fiber placement system with model uncertainties and closed architecture," *IEEE/ASME Transactions on Mechatronics*. (Submitted)

• H. Shen, W.-F. Xie, and **N. Zhu**, "Degeneracy-aware full-pose path planning strategy for robot manipulator," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, pp. 1-11, 2024.

• **N. Zhu**, W.-F. Xie, and H. Shen, "Trajectory planning of cooperative robotic system for automated fiber placement in a leader-follower formation," *International Journal of Advanced Manufacturing Technology*, vol. 130, no. 1, pp. 575–588, 2024.

• **N. Zhu**, W.-F. Xie, and H. Shen, "Position-based visual servoing of a 6-RSS parallel robot using adaptive sliding mode control," *ISA Transactions*, vol. 144, pp. 398-408, 2024.

Conference Paper

• **N. Zhu**, W.-F. Xie, and O. Toker, "Deep-recurrent-neural-network-based adaptive sliding mode control for a 6-DOF industrial serial robot," *IEEE 22nd International Conference on Industrial Informatics (INDIN)*, 2024. (Accepted)

• **N. Zhu**, W.-F. Xie, and H. Shen, "A leader-follower trajectory planning approach for cooperative robotic system in automated fiber placement," *IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1829-1834, 2023.

• **N. Zhu**, W. Xie, and H. Shen, "Adaptive sliding mode control with RBF neural network-based tuning method for parallel robot," *48th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pp. 1-6, 2022.

1.6 Thesis Organization

The organization of the 7 chapters in this thesis is described as follows. In Chapter 1, a brief introduction to this Ph.D. work is provided. In Chapter 2, the literature review results regarding the trajectory planning and control of cooperative robotic systems are presented. The experimental setup of the cooperative robotic system and a distributed control structure with an event-triggered communication network are introduced in Chapter 3. In Chapter 4, a leader-follower trajectory planning scheme for the cooperative robotic

system is explained. A PBVS-based adaptive sliding mode control approach for the 6-RSS parallel robot is described in Chapter 5. In Chapter 6, a distributed DRNN-based adaptive sliding mode control strategy for the cooperative robotic system is illustrated. Conclusion and potential future work are given in Chapter 7. Simulation and experimental results are presented in Chapters 4-6 to demonstrate the feasibility and advantages of the proposed trajectory planning and control strategy.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, the literature review results on the relevant topics involved in this Ph.D research work are given. The following topics are discussed: cooperative manipulation of multiple robots, control structure of cooperative robotic systems, trajectory planning of cooperative robotic systems, advanced nonlinear control for robotic systems, and visual servoing for robotic systems.

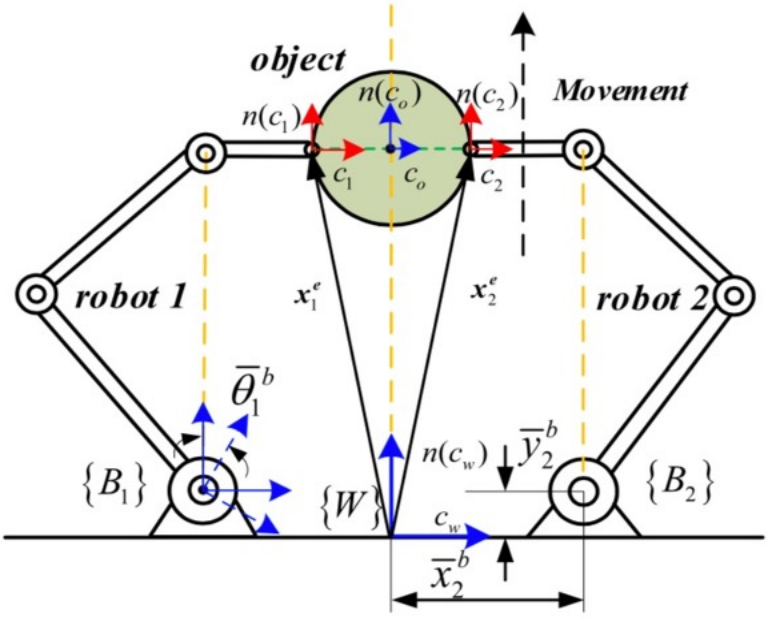
2.2 Cooperative Manipulation of Multiple Robots

Industrial serial robots can perform operations accurately and repeatedly in various manufacturing sectors such as casting, painting, sorting, stacking, welding, and so forth [20]. A serial robot is designed as a series of anthropomorphic links extending from a base platform to an end-effector with multiple motor-actuated joints [21]. Parallel robots have extensive applications in industry such as flight simulators [22], radio telescopes [23], rehabilitation devices [24], vibration isolators [25], motion simulators [26], and manufacturing tools [27]. A parallel robot owns a closed-loop mechanism structure and the end-effector

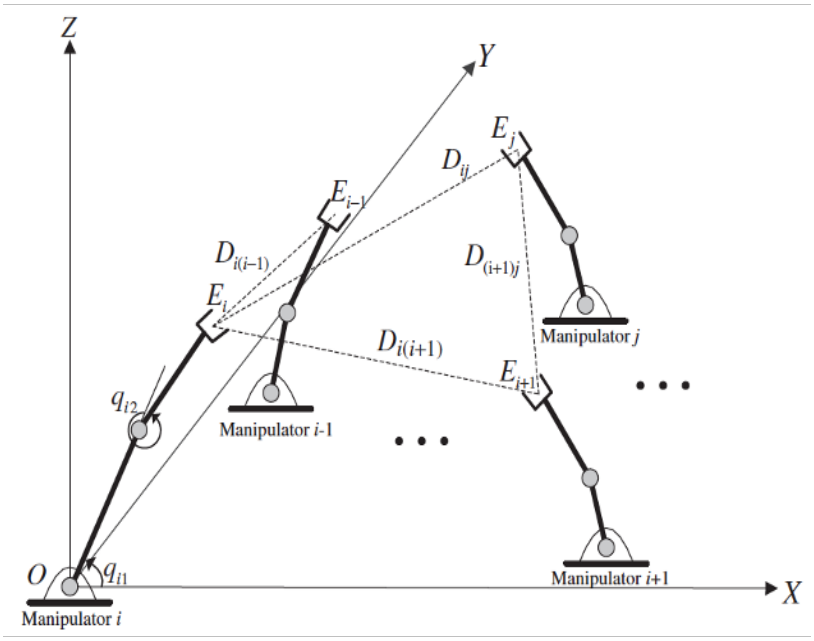
is actuated by multiple serial chains [28]. The dynamic models of high-DOF serial robots and parallel robots are quite nonlinear and coupled with time-varying parameters.

Cooperative manipulation is to maintain certain kinematic relationships among robots during the tracking process [29]. Two cooperative manipulation tasks have been widely investigated in the past decades, as depicted in Figure 2.1. One is that multiple robots collaboratively grab a common object to move along the desired trajectories, the other is that multiple robots track the desired trajectories to achieve a required formation. In [30], the synchronized control of two cooperative robotic manipulators are studied for rigidly grasping a circular disc under base coordinate uncertainties. A distributed cooperative control strategy is proposed in [31] for multirobot systems tracking the desired trajectories with unknown object dynamics, grabbing positions and external disturbances. In [32], a constrained optimization approach is designed for the navigation of a team of mobile manipulators in dynamic environments. The robots can reconfigure the formation to avoid collisions with static and dynamic obstacles while carrying a rigid object. In [33], the robust adaptive synchronization formation control of a group of two-link manipulators is investigated, with the existence of uncertain nonlinearities and parametric uncertainties. A decentralized formation-based iterative learning impedance control architecture for the cooperative control of multiple robot manipulators is presented and discussed in [34]. The mutual synchronization control of multiple robot manipulators with unknown dynamics and external disturbances is considered in [35], under the circumstance that the desired trajectory is only available to part of the team robots.

The AFP manufacturing process in the 13-DOF cooperative robotic system is more complicated than the above-mentioned tasks for the following reasons: (1) Most of the previous work deals with a group of homogeneous robots, while the cooperative AFP system is composed of two different types of robots (serial robot and parallel robot). (2) The two robots hold two different objects (AFP head and Y-shape mandrel). How to satisfy the



(a)



(b)

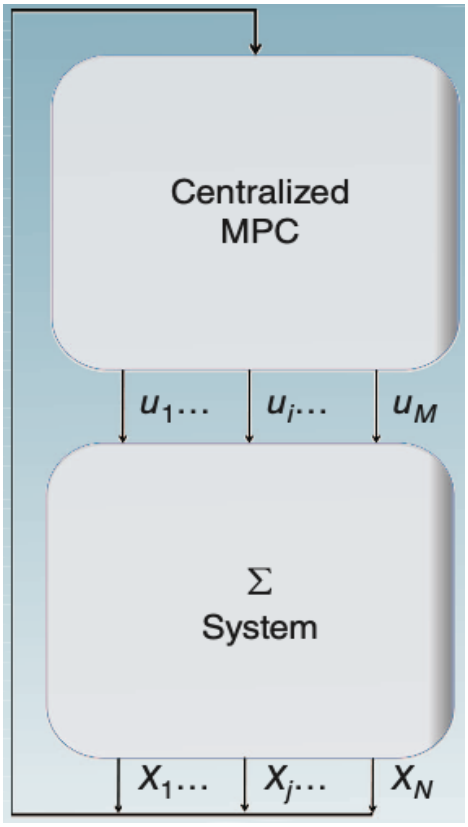
Figure 2.1: Two cooperative manipulation tasks. (a) Grabbing a common object [30]. (b) Achieving a required formation [33].

tracking performance requirements in control of industrial robots has become a sophisticated issue. Therefore, the trajectory planning and control of the cooperative AFP system is worth investigating and needs to be addressed.

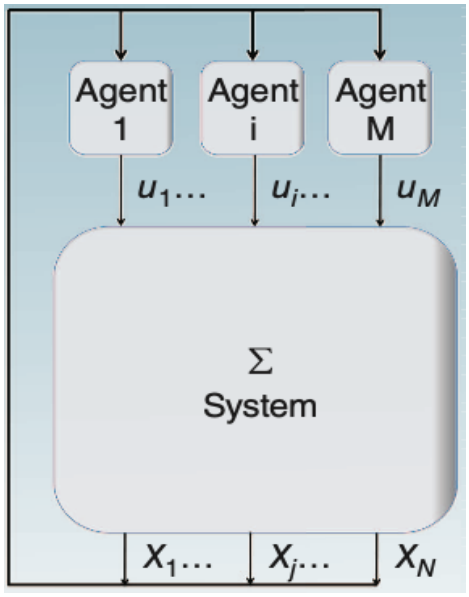
2.3 Control Structure of Cooperative Robotic Systems

The endeavor to accomplish cooperative manipulation by multiple robots normally falls into three categories [36–38]: centralized control, decentralized control and distributed control. A brief demonstration of the three control structures with model predictive control is presented in Figure 2.2. Centralized control systems operate from a central control unit and exhibit exceptional performance. Nevertheless, they involve increased communication requirements and are vulnerable to single-point failures and external attacks. In contrast, in decentralized control, every agent calculates its own control signal and is actuated by the individual controller. The action decisions are made without exchanging information among each other, which may lead to a less performant system. Enabling a certain level of communication among agents can enhance the performance of decentralized control, leading to distributed control. It can avoid complicated communication structures and guarantee the desired cooperation performance with only limited interacted correspondence.

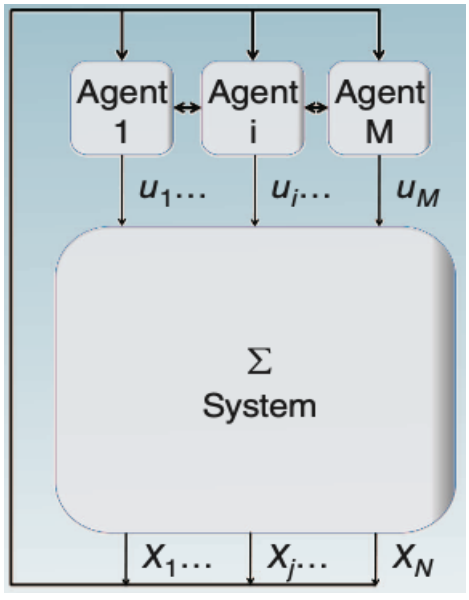
In practice, high communication workload among multiple agents may impair system performance and destabilize the entire network. Event-triggered communication structure [39], where the agents exchange information only upon the fulfillment of specific triggering conditions, has been served as a solution for saving communication and computational resources. The distributed control with event-triggered communication network for cooperative manipulation among multiple homogeneous robots has been explored by some researchers. In [40], a distributed and event-triggered adaptive control strategy is developed for cooperative object manipulation with unknown dynamic parameters and rolling



(a)



(b)



(c)

Figure 2.2: Different control architectures based on model predictive control [37]. (a) Centralized control. (b) Decentralized control. (c) Distributed control. $X_1, ..., X_N$ and $u_1, ..., u_M$ represent system states and actions, respectively.

contacts. A distributed impedance control algorithm with flexible control design is proposed for cooperative object grasping task in [38], and an event-triggered update method is designed to decrease the communication rates. In [41], the adaptive fault tolerant control incorporating collision avoidance is studied for a team of wheel mobile robots, using distributed state estimators and an event-triggered network with communication constraints. In [42], a fully distributed event-triggered control network is designed for the formation control of multiple robots, addressing the problems of dynamic coupling, limited velocity information, and network transmission load. The proposed cooperative control approaches in [38, 40–42] have been validated by simulation and experiment. Nonetheless, the topic regarding distributed control structure with event-triggered conditions remains unresolved, within the domain of cooperative manipulation with two different kinds of robots, i.e. the 13-DOF AFP system.

2.4 Trajectory Planning of Cooperative Robotic Systems

For the purpose of executing the robot motion, trajectory planning aims to generate the reference inputs for the robot control system. The geometric path and the kinematic and dynamic constraints of the robots serve as the inputs to the trajectory planning method. The output is the desired joint or end-effector trajectory, which is expressed as a time sequence of position, velocity and acceleration [43].

Trajectory planning of robotic systems based on optimal control has attracted great attention. Optimal control is to determine the inputs to a dynamic system that minimize a performance metric and satisfy the motion constraints [44]. Execution time is important for the applications of industrial robots, thus lots of trajectory planning approaches focus on generating time-optimal trajectory. Besides, some researchers work on trajectory planning based on time-jerk optimality by adding a regularization term to penalize jerk in the cost function. Limiting joint jerks is essential since the excessive jerks can increase actuator

wear, cause vibrations, and deteriorate tracking speed and accuracy, which will result in non-smooth robot motion [45].

In [46], for a 4-DOF parallel manipulator with a four-to-two leg structure, a time-optimal trajectory planning method is developed. It improves the motion performance by simplifying the kinematics analysis and reducing the kinematics nonlinearity. A time-optimal trajectory planning scheme is designed for a hyper-redundant manipulator in [47]. It generates an end-effector trajectory for the manipulator considering the presence of obstacles and the velocity and acceleration constraints. In [48], a convex time-optimal trajectory planning approach incorporating jerk constraints is proposed to efficiently generate smooth trajectories for robotic manipulators. Considering the execution time and the squared jerk in two objective functions, an optimal trajectory planning method is presented in [49] for smooth trajectory generation of robot manipulators based on multiquadric radial basis functions. In [50], multi-objective non-dominated sorting genetic algorithm is utilized to cope with time-jerk trajectory optimization problem with nonlinear constraints for robot manipulators. In [51], a path-constrained and collision-free time-jerk optimal trajectory planning algorithm is developed for articulated robots in the presence of obstacles and kinematic and dynamic constraints of the robot.

Leader-follower control has been broadly studied for the tracking control of cooperative multi-agent systems. The followers have access to the measurement information of the leader, and the continuous tracking control algorithms are designed for all the robots [52, 53]. In the presence of parameter uncertainties, actuator failures, and uncertain time-varying boundary disturbances, a leader-follower consensus control scheme is implemented for a network composed of multiple flexible manipulators in [54]. With the undirected interconnection graph and variable interconnection time-delays, [55] investigates the leader–follower problem for a team of nonidentical flexible-joint robot manipulators. In [56], the trajectory tracking control of a 4-DOF leader–follower robotic manipulator

system is solved based on a grey-box modelling approach and a fuzzy logic controller. In [57], a synchronized control strategy based on RBF neural network and adaptive control is developed, for multiple robotic manipulators utilizing the leader–follower communication topology.

Regarding the cooperative manipulation among multiple robots, the trajectories of the followers can be determined according to the trajectory of the leader to maintain the required kinematic relationships. For the 13-DOF cooperative robotic system in this project, with the existence of the motion constraints of the robots, how to regulate the trajectories of the serial robot and parallel robot to accomplish the complex AFP task should be considered.

2.5 Advanced Nonlinear Control for Robotic Systems

To eliminate the discrepancies between the nominal dynamic model and the real robotic system for enhancing the tracking control performance, many advanced nonlinear control techniques considering robot dynamics have been developed.

Sliding mode control is recognized for its exceptional robustness against parameter uncertainties and external disturbances in robot control [58]. Neural networks are known with powerful mapping and learning abilities to approximate any nonlinear function, which can effectively address the dynamic uncertainties within model-based nonlinear controllers for robotic systems [59]. Typically, one-hidden-layer shallow feedforward neural networks require substantial neurons for adequate approximation of complex functions. This necessity can result in long training time and high computational complexity. In contrast, deep neural networks with multiple hidden layers can exhibit superior approximation capabilities with less computational consumption, especially for complicated dynamic systems with high DOFs [60]. Feedforward neural networks with multi-layer perceptrons normally lack

inherent dynamic properties. On the contrary, recurrent neural networks incorporate additional feedback loops to store and process the dynamic information of the previous steps, endowing them with stronger dynamic estimation and representation capabilities [61].

In [62], a robust centralized control scheme based on sliding mode control and model predictive control is presented, for impedance control and reference tracking of redundant robotic manipulators in operational space. In [63], a robust cascaded vision and force control approach is designed for industrial robots with model uncertainties and unknown workpiece interaction. A robust continuous integral sliding mode controller is proposed using modified supertwisting algorithm for reducing the chattering problem. In [64], an adaptive incremental sliding mode control approach, including a nominal controller and an sliding mode controller based on positive semi-definite barrier function, is developed for a robot manipulator. A second-order-sliding-mode-based synchronization control scheme is proposed for cable-driven parallel robots in [65]. It aims to guarantee desired trajectory tracking performance and improve the synchronization motion relationship among all the cables, in the presence of model uncertainties and external disturbances. A robust finite-time non-singular terminal sliding mode controller for the trajectory tracking of robot manipulators with full-state constraints is introduced in [66]. A time-varying gain is designed to solve the singularity problem in the classical terminal sliding mode controller.

Taking into consideration control input saturation and output error constraints, a trajectory tracking controller for the robot manipulator is presented in [67]. A simplified RBF neural network structure is employed to approximate the lumped uncertainties of the robot. In [68], an adaptive admittance control method is proposed for robots to interact with time-varying environments to guarantee the trajectory tracking performance. An RBF neural network with a dynamic learning framework is developed to deal with dynamic uncertainties of the system. In order to minimize the position tracking errors for a five-bar parallel robot, the differential evolution and a deep neural network are adopted in [69]

to find the optimal control gains in a cascaded proportional-derivative (PI) controller. In [70], the adaptive neural trajectory tracking control with bounded inputs is investigated for cable-driven parallel robots, and an adaptive multi-layer neural network is used to compensate the modeling uncertainties in the system. In [71], a deep convolutional neural network structure is designed to identify the tool dynamics in bilateral teleoperation. It can achieve accurate force sensing with fast computation and noise robustness. In [72], the motion-force control problem for redundant manipulators considering physical constraints and torque optimization is studied, and a recurrent neural network is established to solve the modified quadratic-programming problem. In [73], a neural-learning Cartesian admittance control scheme is proposed for robotic manipulator control. An improved recurrent neural network is designed to solve the multi-task optimization problem using remote center of motion constraints.

Recent decades have witnessed the integration of above control techniques for the control of robotic systems. In [74], an RBF-neural-network-based sliding mode controller is designed for an uncertain robot. The switching gain is presented as a dynamic model approximated value, which is generated through the adaptation to the unknown dynamics and disturbances using the neural network. A neural-network-based sliding mode control scheme for uncertain industrial robotic manipulator systems with switching loads is developed in [75]. An RBF neural network is applied for approaching to the plant and addresses the limitation of the accurate model of the robotic system. In [76], an adaptive chattering free sliding mode control approach is proposed for the tracking control of redundant parallel manipulators, with an RBF neural network for the estimation of a control term including modeling uncertainties, frictional uncertainties, and external disturbances. In [77], a sliding mode control approach based on neural network is proposed for a 6-DOF robotic manipulator, addressing system uncertainties, input deadzone, and external perturbations. As introduced in [78], for the tracking control of rigid robotic manipulators, a

deep convolutional neural network is utilized to compensate the uncertainties of the system in a fractional-order terminal sliding mode controller, without the prior knowledge of the upper-bounds. In [79], an optimal integral sliding mode controller for mobile robotic manipulators under time-varying system uncertainties and external disturbances is studied. A double feedback recurrent neural network is proposed to approximate nonlinear systems for enhancing the robustness against system uncertainties.

Although sufficient and valuable research results have been obtained, there are still a large number of topics to be investigated for the control of high-DOF robotic systems with the combination of sliding mode control techniques and various neural networks. In this project, some novel control algorithms regarding this topic will be studied for the 13-DOF cooperative robotic system.

2.6 Visual Servoing for Robotic Systems

With the increasing demands of highly accurate pose measurements of the robots, the laser-based and the vision-based tracking systems have been extensively applied in industry. In [80], an autonomous welding strategy using laser structure light is proposed to enhance the accuracy of recognizing the complex spatial weld seam. In [81], the weld seam positioning accuracy in robotic arc welding is increased with the measurements of laser and vision sensors. Nevertheless, the laser-based tracking technique relies on the laser tracker with high cost and requires a large and open workspace. In contrast, pure vision-based system can measure the robot pose using low-cost cameras. By employing the visual guidance technique, an error compensation method is developed to improve the pose accuracy of industrial robots in [82]. In [83], a monocular-vision-based pose estimation system using long short-term memory neural network and sparse regression for robotic machining is presented.

Visual servoing is effective for enhancing the flexibility and accuracy of robotic system.

The robot is controlled to move from the current pose to the desired pose using the visual measurement information. There are three kinds of visual servoing schemes: position-based visual servoing, image-based visual servoing (IBVS), and hybrid visual servoing. Based on different positions of the vision sensor, the configuration of visual servoing system is classified as eye-in-hand and eye-to-hand. In the eye-in-hand configuration, the camera is mounted on the end-effector of the manipulator, so it is only able to observe the target. The prior information of the pose transformation between the camera and the end-effector should be known for the estimation process. In the eye-to-hand configuration, the camera is located in the workspace of the manipulator as a global sensor. It can observe the target and the environment simultaneously, and the image information and the robot motion are completely decoupled.

In PBVS, the control error is represented as the difference between the desired pose and the current pose of the target in 3D Cartesian space. In [84], an acceleration-based controller is designed within a second-order PBVS framework for a robot manipulator. In [85], a PBVS strategy using instantaneous inverse kinematic model is developed for a 6-DOF cable-driven parallel robot to displace heavy payloads. [86] proposes a PBVS control system based on dual gradient neural dynamic models for the position and orientation control of dual robotic arms. For IBVS, the control law is computed based on the image features in 2D image space. In [87], an adaptive switch IBVS controller for industrial robots is developed. The translational motion and the rotational motion of the camera are decoupled, and the IBVS control is decomposed into three separate stages with various gains. In [88], an IBVS method for robot regulation using an uncalibrated eye-to-hand camera is proposed, and the composite learning mechanism is integrated to improve parameter convergence of the camera. In [89], a motion sensing system using a 6-Spherical-Prismatic-Spherical (6-SPS) Stewart platform based on IBVS is established and tested in real-time experiment. In terms of hybrid visual servoing, it utilizes the 2D and 3D features simultaneously for pose

estimation. In [90], a trajectory-tracking-based control approach is proposed for a 6-DOF cable-driven parallel robot to improve the robustness of the hybrid visual servoing, and a tension correction algorithm is developed further in [91] to deal with cable slackness. In [92], a hybrid visual servoing approach is designed for omnidirectional mobile manipulator using a single mounted camera. It can guarantee the system robustness with the appearance of kinematic uncertainties.

For PBVS, the camera model and target geometric model should be known as the prior information. It can realize the pose estimation of the target in Cartesian space, which is important for industrial applications requiring accurate end-effector trajectory tracking. IBVS is robust to camera calibration errors and target modeling errors, but it encounters the problems of singularity and local minima of the image Jacobian matrix [93]. Although hybrid visual servoing combines the advantages of PBVS and IBVS, it suffers from high computational complexity and is more sensitive to image noises. In this project, a photogrammetry sensor C-Track 780 in the eye-to-hand configuration is able to measure the real-time pose of the robot end-effector, which makes it convenient and flexible to apply PBVS for trajectory tracking.

2.7 Summary

A comprehensive literature review on the trajectory planning and control of cooperative robotic systems is conducted in this chapter. First, current trends of cooperative manipulation tasks are introduced. Most of the previous research focuses on either common object grasping or formation control of multiple robots. Then, different control structures for cooperative robotic systems, including centralized control, decentralized control and distributed control are described. Especially, distributed control with event-triggered communication network is analyzed. Besides, the introduction to optimal-control-based trajectory planning technique and leader-follower control of multiple robots is given. In addition,

advanced nonlinear control techniques for robotic systems, consisting of sliding mode control and neural network control are presented. Finally, the widely investigated robotic visual servoing techniques are introduced.

Chapter 3

Distributed Control of the Cooperative Robotic System

3.1 Introduction

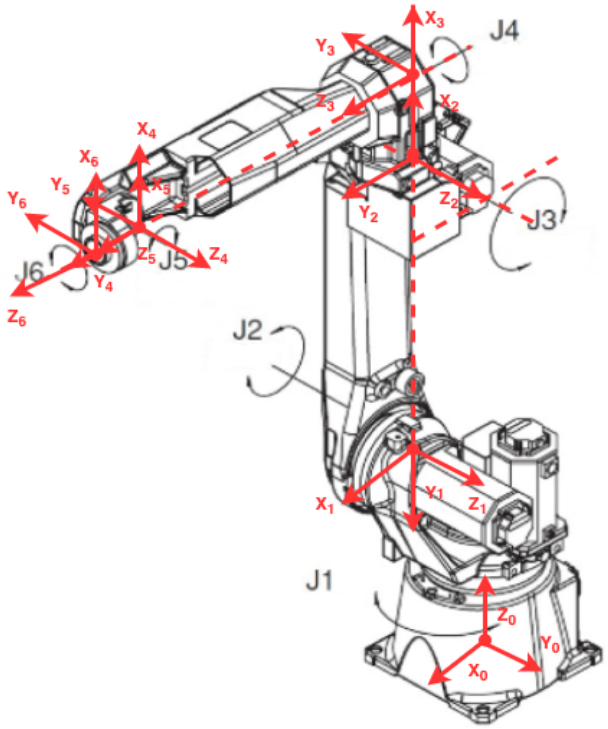
In this chapter, the experimental setup of the 13-DOF cooperative robotic system with the control network is introduced. For the effective cooperative control of the AFP system, a distributed control structure with event-triggered condition is proposed. Different from [38, 40–42], where the control of multiple homogeneous robots is investigated, the serial robot and the parallel robot in the system are controlled by individual controllers. The photogrammetry sensor C-Track 780 in the eye-to-hand configuration is applied to accurately measure the 3D target pose. Therefore, an event-triggered communication network with low workload and limited information exchange, is established based on the measured end-effector pose of the serial robot in real time. The principle of the vision-based pose estimation is also described in this chapter.

The uncertain noises in robotic visual measurements may lower the pose estimation accuracy. Kalman filter (KF) has been widely utilized for localization, navigation, tracking, perception, estimation, and motion control [94]. It can obtain the estimation results close

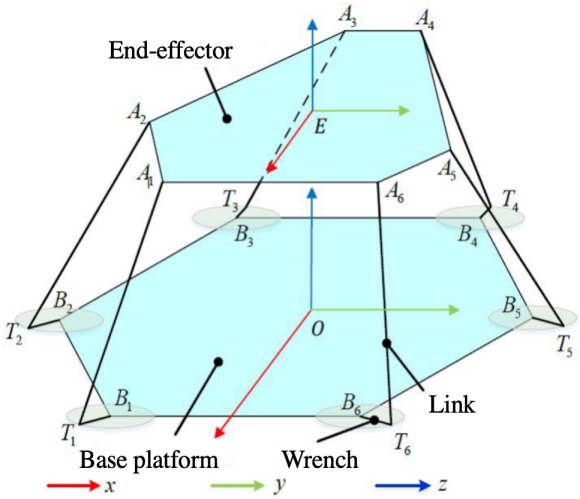
to the true values with visual measurements containing noises and uncertainties. While the measurement noise covariance matrix can be predetermined by experiments, the process noise covariance matrix should be tuned with the change of the robot motion speed [95]. However, the dynamics of the robot motion regarding the vision sensor may not be predicted accurately, which makes it difficult to tune the process noise covariance matrix. Insufficient noise estimates probably cause the filter divergence and deteriorate the system performance [96]. To address this issue, an adaptive Kalman filter is employed to update the covariance matrix of the process noise in this project.

3.2 Experimental Setup of the Cooperative AFP System

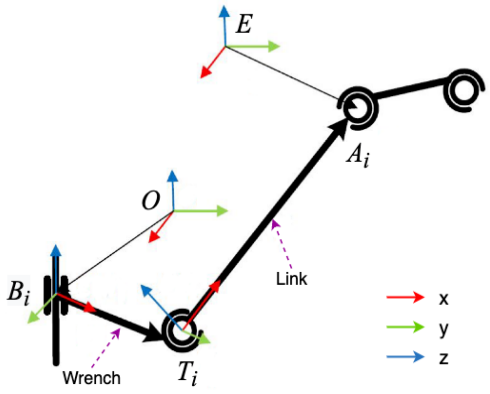
The experimental setup of the cooperative robotic system for AFP has been demonstrated in Figure 1.4. The 6-DOF serial robot is Fanuc M-20iA, a wrist-partitioned articulated robotic manipulator with six revolute joints. The 6-RSS parallel robot is Model 710LP-6-500-220 provided by Servo & Simulation Inc. The base platform is connected with the end-effector through six serial chains. Each serial chain is composed of a link and a wrench. The end-effector motion is actuated by six revolute joints, while twelve spherical joints are used to concatenate the links with the end-effector and the wrenches, respectively. The schematics of the two robots are presented in Figure 3.1. The 1-DOF rotary stage is AGR75-NC-9DU-BMS-R-3 provided by Aerotech Inc. One three-jaw chuck is attached to the rotary stage to hold the Y-shape mandrel. The AFP head shown in Figure 3.2 includes four main subsystems: tow tensioning, feeding and cutting parts, and compaction roller [97]. The compaction roller is utilized to compress and adhere the fiber to the mandrel surface. The photogrammetry sensor C-Track 780 is provided by Creaform Inc. It is a dual camera with the ability of tracking a set of reflectors for real-time pose estimation. The maximum measuring volume is 7.8 m^3, and the sampling frequency is 29 Hz. The two robots and AFP head can be modeled with frame definitions in the software VXelements,



(a)



(b)



(c)

Figure 3.1: Schematics of the two robots in the AFP system. (a) Fanuc M-20iA robot. (b) 6-RSS parallel robot. (c) Single serial chain of the parallel robot.

and some reflectors are attached on them as the feature points. The pose of the frames defined in Figure 1.4 w.r.t. the sensor frame F_s can be measured and tracked in real time.

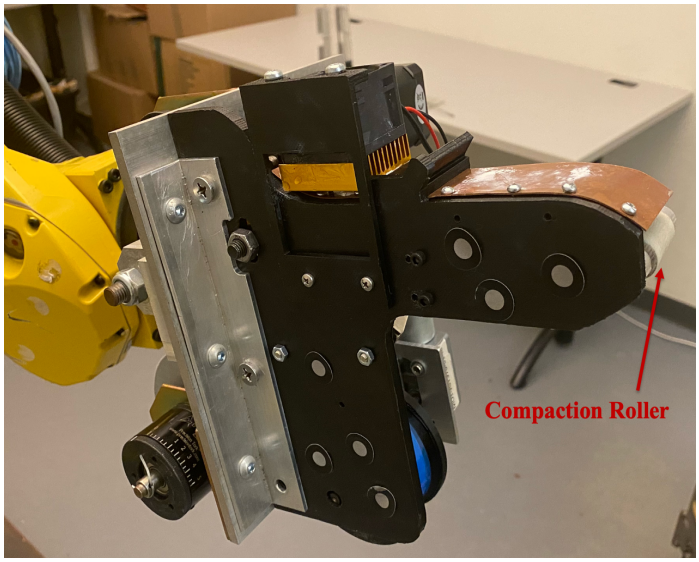


Figure 3.2: AFP head.

The control network of the cooperative AFP system is presented in Figure 3.3. One computer is adopted for the control of the serial robot and C-Track 780, the other computer is employed to control the parallel robot. The real-time communications between the two computers are established using a DTech RS232 USB to DB9 female serial port cable. The information of the end-effector pose of the two robots measured by C-Track 780 can be transmitted from Computer 1 to Computer 2.

3.3 Distributed Control with Event-triggered Communication Network

Regarding the 0° fiber path on the Y-shape mandrel in Figure 1.5, it has been proved in [14] that the AFP head can place fiber on Branch 1 smoothly by just following the desired fiber path. However, for Branch 2 and Branch 3, due to the kinematic and dynamic constraints of the serial robot, the cooperation of two robots is needed for the accomplishment of AFP.

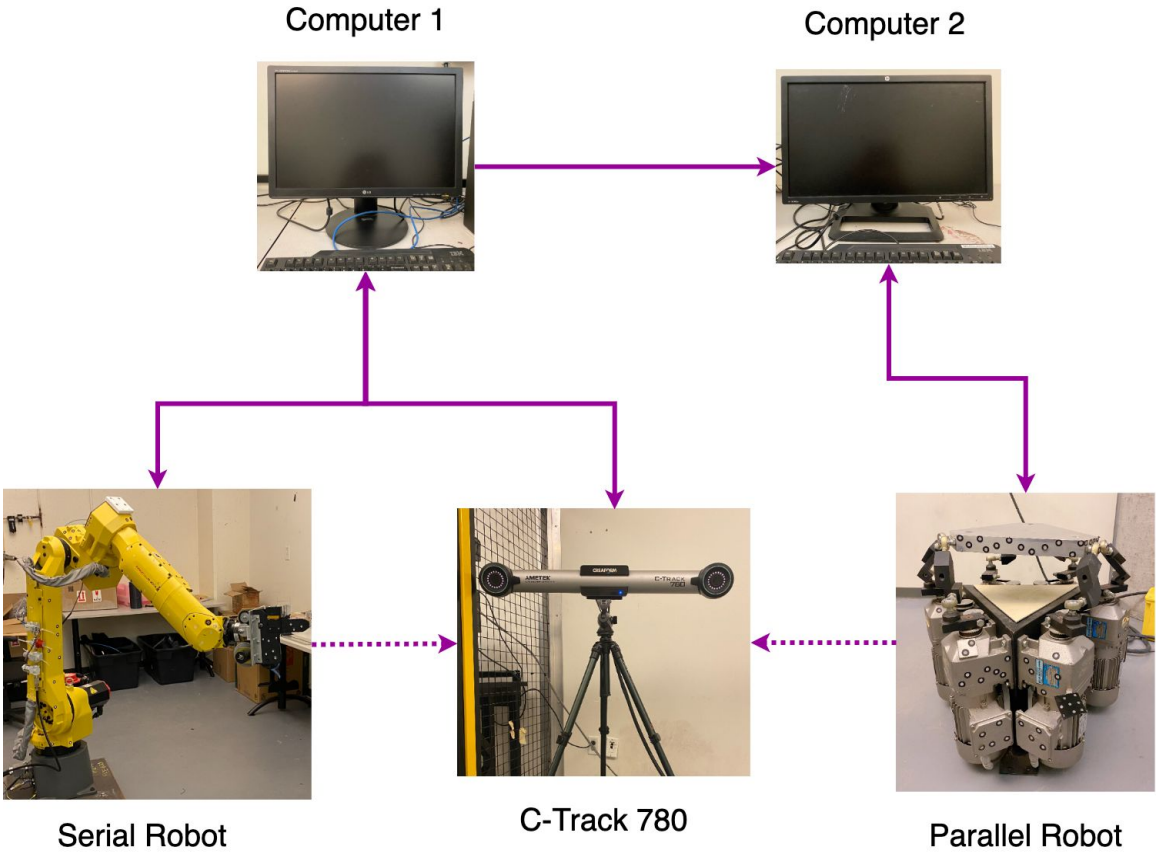


Figure 3.3: Control network of the cooperative AFP system.

The proposed distributed control structure with event-triggered condition is indicated in Figure 3.4. In this project, the trajectory planning and tracking control of the two robots for a continuous fiber layup process on Branches 1 and 2 with 0° fiber path is investigated. The desired trajectories can be pre-planned offline, and individual controllers are applied for the control of the two robots. As for Branch 1, the serial robot can successfully move along the intended fiber path while the parallel robot keeps stationary. The cooperative motion of the parallel robot should be executed for Branch 2. As the AFP head reaches the start point of the fiber path on Branch 2, the end-effector pose information of the serial robot can be received by the photogrammetry sensor C-Track 780. Then the triggering condition for the parallel robot movement is satisfied, a triggering signal will be sent from Computer 1 to Computer 2 to start the motion of the parallel robot.

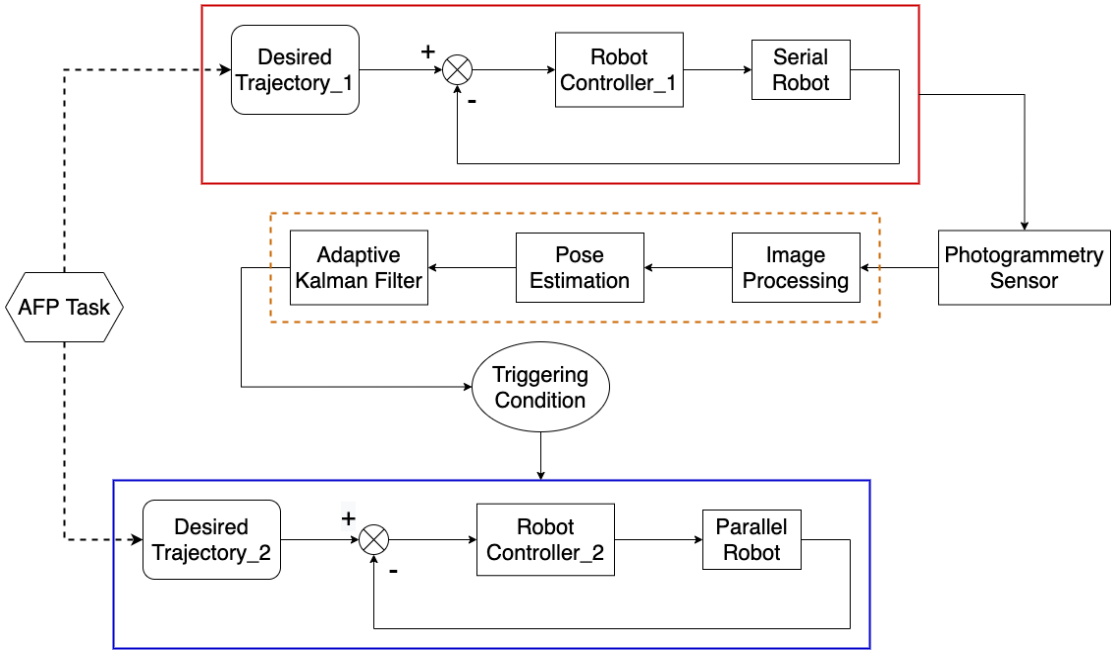


Figure 3.4: Distributed control structure of the cooperative robotic system.

3.4 Pose Estimation

The photogrammetry sensor C-Track 780 can measure the pose of a target with a series of reflectors serving as the feature points. In this section, the pose estimation principle is explained for measuring the end-effector pose of the 6-DOF serial robot. The objective is to measure the pose of F_{es} w.r.t. F_s in Figure 1.4, which is defined as $E^s = [x^s, y^s, z^s, \alpha^s, \beta^s, \gamma^s]^T$, where $[x^s, y^s, z^s]^T$ and $[\alpha^s, \beta^s, \gamma^s]^T$ represent position and orientation, respectively.

Assuming the number of reflectors is n, the homogeneous coordinate of each feature point in F_s is given as ${}^sP_i = (x_i^s, y_i^s, z_i^s, 1)$, $i = 1, 2..., n$, and the homogeneous coordinate of each feature point on the jth image plane is denoted as ${}^mP_{ij} = (u_{ij}, v_{ij}, 1), j = 1, 2$. The projection matrix and camera matrix of the jth camera are defined as ${}^m_jT$ and B_j, respectively, which can be known by calibration process. The perspective projection of

C-Track 780 is described as

$$ {}^{m}P_{ij} = {}^{m}_{j}T {}^{s}P_{i}, \quad {}^{m}_{j}T = B_{j}{}^{m}_{s}T_{j} \tag{3.1} $$

where ${}^{m}_{s}T_{j}$ is the homogeneous transformation matrix between F_s and the jth camera frame.

The homogeneous transformation matrix between F_{es} and F_s is calculated as

$$ {}^{s}_{es}T = \begin{bmatrix} R(\alpha^s, \beta^s, \gamma^s) & (x^s, y^s, z^s)^T \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{3.2} $$

where $R(\alpha^s, \beta^s, \gamma^s)$ is the rotational matrix given as

$$ R = \begin{bmatrix} c\alpha^s c\beta^s & c\alpha^s s\beta^s s\gamma^s - s\alpha^s c\gamma^s & c\alpha^s s\beta^s c\gamma^s + s\alpha^s s\gamma^s \\ s\alpha^s c\beta^s & s\alpha^s s\beta^s s\gamma^s + c\alpha^s c\gamma^s & s\alpha^s s\beta^s c\gamma^s - c\alpha^s s\gamma^s \\ -s\beta^s & c\beta^s s\gamma^s & c\beta^s c\gamma^s \end{bmatrix} \tag{3.3} $$

where $s\alpha^s = \sin(\alpha^s)$ and $c\alpha^s = \cos(\alpha^s)$. The homogeneous coordinate of each feature point in F_{es} can be known as ${}^{es}P_i = (x_i^{es}, y_i^{es}, z_i^{es}, 1)$ after F_{es} is defined. The transformation between ${}^{es}P_i$ and ${}^{s}P_i$ is formulated as

$$ {}^{s}P_i = {}^{s}_{es}T {}^{es}P_i \tag{3.4} $$

where ${}^{s}P_i$ can be computed using Eq. (3.1). According to [98], by selecting more than three non-collinear feature points, a unique solution for E^s can be obtained using Eqs. (3.2)-(3.4). Therefore, the pose of F_{es} w.r.t F_s at different moments can be tracked.

3.5 Adaptive Kalman Filter

An adaptive Kalman filter is utilized to deal with sensor noises in visual measurements to guarantee the pose estimation accuracy. The measurement noise covariance matrix can be determined based on the residual errors of the static measurements from the vision sensor [99], thus the AKF is only applied for updating the covariance matrix of the process noise. The details of this approach are presented as follows.

Let k be the sampling step, the state vector of the serial robot including pose and velocity is given as

$$\Psi_{k,k} = [x(k), y(k), z(k), \alpha(k), \beta(k), \gamma(k), \dot{x}(k), \dot{y}(k), \dot{z}(k), \dot{\alpha}(k), \dot{\beta}(k), \dot{\gamma}(k)]^T. \quad (3.5)$$

The predicted state $\hat{\Psi}_{k,k-1}$ at step k can be calculated based on the estimated state $\hat{\Psi}_{k-1,k-1}$ at step $k-1$, and the prediction of error covariance $P_{k,k-1}$ at step k can also be obtained. The prediction is expressed as

$$\hat{\Psi}_{k,k-1} = A\hat{\Psi}_{k-1,k-1} + W_{k-1} \quad (3.6)$$

$$P_{k,k-1} = AP_{k-1,k-1}A^T + Q_{k-1} \quad (3.7)$$

where $A \in R^{12\times12}$ is the state transition matrix, the diagonal elements of A are all given as 1, $A_{i,i+6} = T$, T is the sampling interval, and $i = 1, 2, ..., 6$. The process noise W_{k-1} is zero-mean Gaussian noise with the covariance matrix Q_{k-1}.

The state measurement equation is given as

$$Z_k = H\Psi_k + V_k \quad (3.8)$$

where $Z_k \in R^6$ is the pose measurement at step k, $H \in R^{6\times12}$ is the measurement transition

matrix, and the measurement noise V_k is zero-mean Gaussian noise with the covariance matrix S_k. In this project, Z_k can be acquired from the C-Track 780 directly, and H_k is defined as $[I_{6*6}, 0_{6*6}]$.

The Kalman gain K_k, estimated state $\hat{\Psi}_{k,k}$ and estimated error covariance $P_{k,k}$ at step k can be updated as follows:

$$K_k = P_{k,k-1}H^T(HP_{k,k-1}H^T + S_k)^{-1} \tag{3.9}$$

$$\hat{\Psi}_{k,k} = \hat{\Psi}_{k,k-1} + K_k(Z_k - H\hat{\Psi}_{k,k-1}) \tag{3.10}$$

$$P_{k,k} = P_{k,k-1} - K_k H P_{k,k-1}. \tag{3.11}$$

According to [96], an adaptation law is performed to update Q_k. The process noise is estimated as

$$\hat{\zeta}_j = \hat{\Psi}_{j,j-1} - A\hat{\Psi}_{j-1,j-1} \tag{3.12}$$

where $j = k - N, ..., k$, and N is the window length of the previous measurements for adaptation. Then one defines

$$\Delta_k = AP_{k-1,k-1}A^T - P_{k,k}. \tag{3.13}$$

The process noise covariance matrix is updated using

$$\bar{\zeta}_k = \bar{\zeta}_{k-1} + \frac{1}{N}(\hat{\zeta}_k - \hat{\zeta}_{k-N}) \tag{3.14}$$

$$\begin{aligned} Q_k = Q_{k-1} + \frac{1}{N-1}\Bigg(& (\hat{\zeta}_k - \bar{\zeta}_k)(\hat{\zeta}_k - \bar{\zeta}_k)^T - (\hat{\zeta}_{k-N} - \bar{\zeta}_k)(\hat{\zeta}_{k-N} - \bar{\zeta}_k)^T \\ & + \frac{N-1}{N}(\Delta_{k-N} - \Delta_k) + \frac{1}{N}(\hat{\zeta}_k - \hat{\zeta}_{k-N})(\hat{\zeta}_k - \hat{\zeta}_{k-N})^T \Bigg). \end{aligned} \tag{3.15}$$

The diagonal elements of Q_k should be set as their absolute values to ensure that Q_k is

positive definite.

3.6 Summary

In this chapter, the hardware setup and control network of the 13-DOF cooperative robotic system are introduced. To maintain the functional collaborative control of the AFP system, a distributed control structure based on an event-triggered communication network is proposed. Two individual controllers are applied to control the serial robot and the parallel robot, respectively. The photogrammetry sensor C-Track 780 is employed to realize the accurate end-effector pose estimation of the serial robot. The information transmission between the two robots starts only when the predefined condition is satisfied. An adaptive Kalman filter is adopted to address the uncertain noises in visual pose estimation.

Chapter 4

Leader-follower Trajectory Planning for the Cooperative Robotic System

4.1 Problem Formulation

The desired trajectory generation for the two robots is crucial for the accomplishment of the AFP task. One main obstacle is that the kinematic and dynamic constraints of the serial robot may disrupt the AFP process, which has severe influence on the final product quality. Trajectory planning of industrial robots with the consideration of time and jerk optimality can guarantee the desired execution time and trajectory smoothness. Taking into account the complicated geometry of the Y-shape mandrel and the desired 0° fiber path, this Ph.D. project aims to design an optimal trajectory planning algorithm in joint space for the serial robot, while subjecting to the motion constraints and maintaining the smooth AFP operational process.

Considering the efficient real-world applications, it is judicious to solve optimal control problems by numerical methods, which are categorized into two types: indirect and direct. Since it is sophisticated to realize the initialization and derivation of the optimal conditions in indirect methods [100], direct methods have been more widely applied. For

direct methods, the continuous-time optimal control problem is converted into a discrete nonlinear program (NLP) by transcription, and a nonlinear optimization solver is implemented to get the optimal solutions [101]. Direct methods are classified into two classes: shooting and collocation. Shooting approaches approximate the system states through a simulation. However, for complicated problems, the nonlinear relationships between the decision variables and the objective functions may not be well approximated using the linear or quadratic models in the NLP solvers [102]. Collocation approaches are much more powerful to cope with optimal control problems. The system states are represented using decision variables at the collocation points directly. When the roots of orthogonal polynomials such as Chebyshev or Legendre are selected as the collocation points, it is known as orthogonal collocation [103]. To address the trajectory optimization of the serial robot, pseudospectral method is adopted to realize the transcription, and interior point method together with automatic differentiation is employed for solving NLP.

The required kinematic relationships should be maintained among robots in cooperative manipulation tasks. As illustrated in [15], by following the planned optimal trajectory, the AFP head path may deviate from the desired fiber path, and the AFP head roller direction may not satisfy the requirement of keeping perpendicular to the mandrel surface. Under this circumstance, the 6-RSS parallel robot should move to compensate the serial robot motion and satisfy the AFP geometric constraints, which can guarantee that fiber is placed with the desired angle 0°. Vision-based tracking systems have seen widespread applications in industry by satisfying the demands of precise pose measurements for robots. In this Ph.D. project, the photogrammetry sensor C-Track 780 is able to measure the pose of a given target with high accuracy and reliability, which can contribute to the trajectory generation of the parallel robot.

4.2 Contributions

Regarding the trajectory planning of the 13-DOF cooperative AFP system, [14] designs a synchronized semi-offline trajectory planning method by adding real-time corrections to the pre-planned trajectory. When the serial robot reaches the singularities or joint constraints, the offset motion will be generated for the parallel robot. In order to maintain certain kinematic relationships with the parallel robot, the compensation motion of the serial robot will be computed, which can free the robot motion from the singularities and joint constraints. The C-Track 780 is adopted to acquire the pose measurements of the two robots for calculating the compensation and offset motion.

This Ph.D. project works on the cooperative trajectory planning from a new perspective. The main contributions are described as follows:

(1) To deal with the cooperative manipulation problem in the 13-DOF AFP system, a leader-follower trajectory planning strategy is proposed. The 6-DOF serial robot with the AFP head is employed as the leader, while the 6-RSS parallel robot holding the Y-shape mandrel through the 1-DOF rotary stage serves as the follower. It is different from the leader-follower formation utilized for the synchronization control of the AFP system in [104].

(2) Considering the robot kinematic and dynamic constraints, a time-jerk optimal trajectory planning scheme is designed for the serial robot. Compared with [14], the trajectory planning in this project is conducted in joint space rather than Cartesian space. The planned joint-space trajectory can effectively avoid the joint singularities, which may occur in the calculation of inverse kinematics if a specific Cartesian-space trajectory is given [51]. In addition, the joint constraints associated with the physical limitations of the robot actuators are also considered.

(3) For the serial robot trajectory optimization, a cost function different from the ones in [49–51, 105] is defined to balance the motion time and the trajectory smoothness. While

B-splines and multiquadric radial basis functions are employed for trajectory interpolation in [49, 50], respectively, pseudospectral method is adopted in our work to realize the transcription. It can increase the solution accuracy using high-order orthogonal polynomials [44] and achieve the exponential convergence rate in the order of the polynomials [103]. Interior point method together with automatic differentiation is implemented for solving NLP. Our work focuses on generating the trajectory with high smoothness and reasonable motion time, while [105] aims to plan the time-optimal trajectory with certain smoothness. Additionally, our work involves the cooperation of two robots rather than just one serial robot, it requires more accurate positioning at the initial and final time and results in narrower reachable workspace of the serial robot end-effector.

(4) A vision-based trajectory generation approach is proposed for the parallel robot. It ensures that fiber can be placed along the desired path by compensating the serial robot motion to satisfy the AFP geometric constraints. With the pose measurements of the C-Track 780, the desired parallel robot trajectory can be planned according to the desired trajectory of the start point on the given fiber path. In [106, 107], the trajectory planning of the parallel robot is conducted based on the robot kinematics and the constant-orientation workspace method, where the 3D workspace can be obtained by tracking the position of a defined point on the motion platform with a constant orientation. However, the visual measurement technique is not involved for improving the positioning accuracy.

(5) Simulation and experimental validations of the proposed trajectory planning strategy have been conducted on the 13-DOF cooperative robotic system, and comparison results demonstrate its superiority over the trajectory planning algorithm in [14].

The organization of the remainder in this chapter is described as follows. At first, the leader-follower trajectory planning strategy is described. Then, the time-jerk optimal trajectory planning scheme for the 6-DOF serial robot is illustrated, and the vision-based

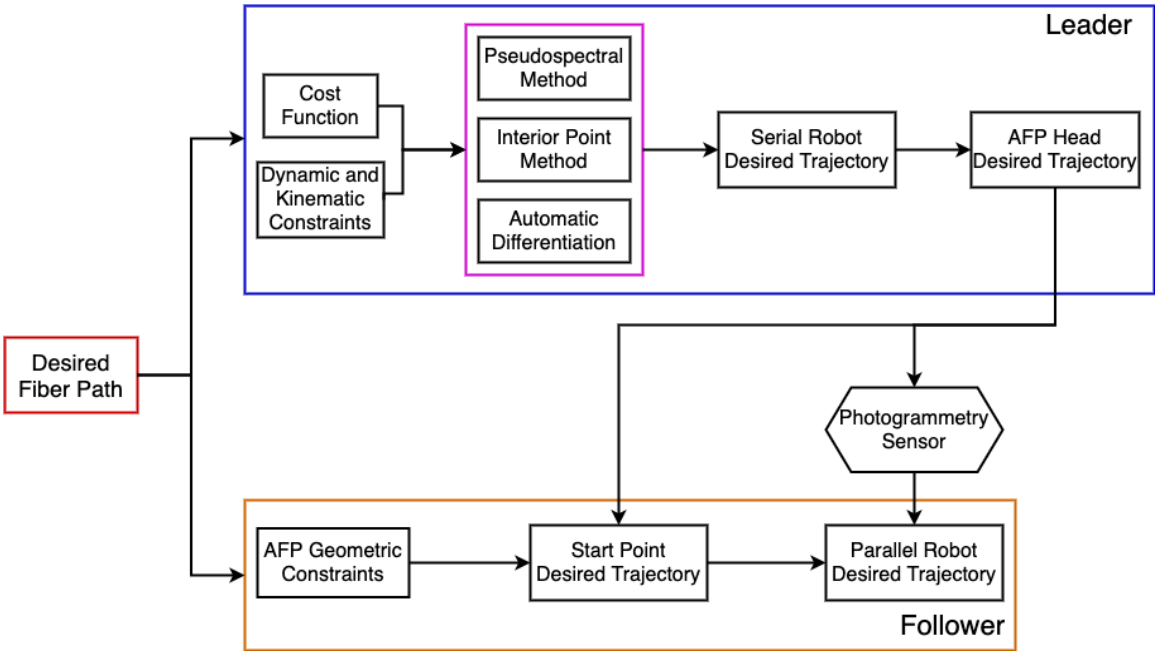


Figure 4.1: Leader-follower trajectory planning strategy.

trajectory generation approach for the 6-RSS parallel robot is introduced. Moreover, simulation and experimental results are demonstrated. The final section concludes the work of this chapter.

4.3 Leader-follower Trajectory Planning

The frame definitions of the 13-DOF cooperative AFP system are shown in Figure 1.4, including serial robot base frame F_{bs} and end-effector frame F_{es}, AFP head roller frame F_r, parallel robot base frame F_{bp} and end-effector frame F_{ep}, and vision sensor frame F_s. Note that the orientation of F_r is defined the same as the orientation of F_{es} along all the three axes, and there is a translation between the origins of the two frames. During the AFP process, the relative pose between F_r and F_{es} will remain constant.

According to [14], the AFP head is able to place fiber along the desired path on Branch 1 without any interruptions. Unfortunately, due to the dynamic and kinematic constraints of

the serial robot, the AFP head cannot move freely along the desired fiber path on Branches 2 and 3, thus the cooperative motion of the two robots is necessary to accomplish the AFP work. The proposed trajectory planning strategy is demonstrated in Figure 4.1. The serial robot works as the leader and the parallel robot works as the follower. A time-jerk optimal trajectory planning scheme is proposed for the serial robot. To compensate the motion of the serial robot, a vision-based trajectory generation method is designed for the parallel robot based on the geometric constraints of AFP. In the following sections, the details of the trajectory planning strategy are presented.

4.4 Trajectory Planning of the 6-DOF Serial Robot

4.4.1 Time-jerk Optimal Trajectory Planning Problem

The dynamics of the 6-DOF serial robot is represented as [10]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_f = \tau \tag{4.1}$$

where $q \in R^6$, $\dot{q} \in R^6$ and $\ddot{q} \in R^6$ are position, velocity and acceleration of the robot joints, respectively. $M(q) \in R^{6\times6}$, $C(q,\dot{q}) \in R^{6\times6}$, $G(q) \in R^6$, $\tau_f \in R^6$, and $\tau \in R^6$ represent inertia matrix, Coriolis and centrifugal matrix, gravitational term, frictional term, and control input, respectively.

Let the system state be $X = [X_1^T, X_2^T]^T$, where $X_1 = q$ and $X_2 = \dot{q}$, and control input $U = \tau$. Trajectory optimization is a technique to get an open-loop solution to an optimal control problem. Given the initial and final joint states, by adjusting the decision variables including final time t_f, state sequence X, as well as control sequence U, the robot should be able to move from the start point to the end point freely and smoothly, while satisfying the motion constraints and minimizing a cost function J.

The cost function J has great influence on the optimization performance and determines the robot motion quality. Regarding the AFP manufacturing, it is a wise choice to define a cost function consisting of a term related to the final time t_f and a regularization term related to the trajectory smoothness. To achieve the time-jerk optimality, the cost function in this work is established as

$$J = \lambda t_f + (1-\lambda)\int_0^{t_f} \dddot{q}^T(t) R \dddot{q}(t) dt \tag{4.2}$$

where $0 < \lambda < 1$ is the weight, $\dddot{q}(t)$ is the joint jerk, and R is a positive diagonal matrix. Minimizing joint jerks can reduce actuator wear and vibration, and guarantee high tracking accuracy.

The dynamic and kinematic constraints of the 6-DOF serial robot are introduced as follows, where $i = 1, 2, ..., 6$.

(a) System dynamics

To describe how the system changes through time, a group of differential equations can be formulated for modeling the nonlinear dynamics of the robot. The dynamic model of the robot is rewritten as

$$\begin{cases} \dot{X}_1 = X_2 \\ \dot{X}_2 = M^{-1}(X_1)\left(U - C\left(X_1, X_2\right) X_2 - G\left(X_1\right) - \tau_f\right) \end{cases} =: F. \tag{4.3}$$

Therefore, the dynamics of the robot can be represented as

$$\dot{X}(t) = F(X(t), U(t)). \tag{4.4}$$

(b) Path constraints

To enforce the restrictions along the trajectory, the path constraints in terms of position,

velocity, torque and torque rate of each joint are assigned as

$$\begin{aligned} \text{Position}: &\quad q_i^{min} \leq q_i(t) \leq q_i^{max}, \\ \text{Velocity}: &\quad \dot{q}_i^{min} \leq \dot{q}_i(t) \leq \dot{q}_i^{max}, \\ \text{Torque}: &\quad \tau_i^{min} \leq \tau_i(t) \leq \tau_i^{max}, \\ \text{Torque rate}: &\quad \dot{\tau}_i^{min} \leq \dot{\tau}_i(t) \leq \dot{\tau}_i^{max}. \end{aligned} \tag{4.5}$$

(c) Boundary constraints

Assuming the robot is stationary at the initial and final time, the boundary constraints that impose restrictions on the initial and final states, including position, velocity and acceleration are given as

$$\begin{aligned} \text{Position}: &\quad q_i(0) = q_i^0, q_i(t_f) = q_i^f, \\ \text{Velocity}: &\quad \dot{q}_i(0) = 0, \dot{q}_i(t_f) = 0, \\ \text{Acceleration}: &\quad \ddot{q}_i(0) = 0, \ddot{q}_i(t_f) = 0, \end{aligned} \tag{4.6}$$

where q_i^0 and q_i^f represent the joint positions at the initial and final time, respectively.

4.4.2 Trajectory Optimization Scheme

Direct method is employed to deal with the trajectory optimization of the serial robot. It mainly includes two aspects: transcription and optimization. First, the continuous-time optimal control problem is discretized into a nonlinear program through transcription. In general, the state and control sequences are discretized along the time slot and represented using the values at the discrete points or knots. Second, a specific nonlinear optimization solver is utilized to solve NLP to get the optimal results. Pseudospectral method is adopted to realize the transcription in this work, and the implementation of this method is via Chebfun [108] in MATLAB. The details of pseudospectral method are introduced as

follows.

(a) Knots calculation

Regarding optimal control, the functions in the time domain $t \in [0, t_f]$ need to be mapped to the interval $\xi \in [-1, 1]$. The mapping can be done by

$$\xi = 2\frac{t}{t_f} - 1. \tag{4.7}$$

The interpolation points are often defined as the roots of orthogonal polynomials. The Legendre orthogonal polynomials are popularly utilized, but there are no closed-form solutions for the roots and they must be calculated numerically [103], while the roots of the Chebyshev orthogonal polynomials are easier to compute. In this work, the Chebyshev-Lobatto points are chosen as the interpolation points, given as

$$\xi_i = \cos\left(\frac{\pi i}{N}\right), \tag{4.8}$$

where $i = 0, 1, ..., N$, and the number of interpolation points is defined as $N + 1$.

In review of Eqs. (4.7) and (4.8), the interpolation knots for $t \in [0, t_f]$ can be calculated as

$$K_i = \frac{t_f}{2}\left(\cos\left(\frac{\pi i}{N}\right) + 1\right). \tag{4.9}$$

In the transcription, $X(t)$ and $U(t)$ will be discretized at different knots as $\{X(K_i), i = 0, 1, ..., N\}$ and $\{U(K_i), i = 0, 1, ..., N\}$, respectively.

(b) Interpolation

Barycentric Lagrange interpolation [109] is an efficient way to evaluate high-order polynomials. The function value at each point is represented by a weighed combination of the function's value at the interpolation knots. The system state and control sequences

can be formulated as

$$\begin{aligned} X_j(t) &= \sum_{i=0}^{N} X_j(K_i)\mathcal{L}_i(t), \\ U_j(t) &= \sum_{i=0}^{N} U_j(K_i)\mathcal{L}_i(t), \end{aligned} \tag{4.10}$$

where $j = 1, 2, ..., 6$, and $\mathcal{L}_i(t)$ is the ith Lagrange polynomial as illustrated in [109], which is given as

$$\begin{aligned} \mathcal{L}_i(t) &= \prod_{n\neq i}(t - K_n) \Big/ \prod_{n\neq i}(K_i - K_n) \\ &= \prod_{n=0}^{N}(t - K_n) \Big/ \left((t - K_i) \prod_{n\neq i}(K_i - K_n) \right), t \neq K_i. \end{aligned} \tag{4.11}$$

If $t = K_i$, then $\mathcal{L}_i(K_i) = 1$, otherwise the interpolation weights μ_i can be described as

$$\mu_i = \frac{1}{\prod_{n\neq i}(K_i - K_n)}. \tag{4.12}$$

By defining $\mathcal{L}(t) = \prod_{n=0}^{N}(t - K_n)$, then Eq. (4.11) can be formulated as

$$\mathcal{L}_i(t) = \frac{\mathcal{L}(t)\mu_i}{t - K_i}. \tag{4.13}$$

The interpolation will be adopted to obtain the solution to the continuous-time optimal control problem after the discrete NLP has been solved.

(c) Differentiation

Orthogonal polynomials are easy to differentiate. The calculated differentiation matrix can establish a series of constraints to enforce the system dynamics. The state and control

sequences at different interpolation knots can be integrated and expressed as

$$\tilde{X}_j = \begin{bmatrix} X_j(K_0) \\ \vdots \\ X_j(K_N) \end{bmatrix}, \quad \tilde{U}_j = \begin{bmatrix} U_j(K_0) \\ \vdots \\ U_j(K_N) \end{bmatrix}. \tag{4.14}$$

The derivatives of Eq. (4.14) are calculated as

$$\frac{d\tilde{X}_j}{dK} = \frac{2}{t_f} D_x \tilde{X}_j, \quad \frac{d\tilde{U}_j}{dK} = \frac{2}{t_f} D_u \tilde{U}_j \tag{4.15}$$

where D_x and D_u are the differential matrices introduced in [109] for computing the derivatives of $\tilde{X}_j$ and $\tilde{U}_j$, respectively. The element in the kth row and mth column of the differentiation matrix can be calculated as

$$D_{km} = \begin{cases} \dfrac{\mu_m/\mu_k}{\xi_k - \xi_m}, & k \neq m, \\ -\displaystyle\sum_{k \neq m} D_{km}, & k = m. \end{cases} \tag{4.16}$$

(d) Integration

Since the Chebyshev-Lobatto points are employed, the Clenshaw-Curtis quadrature rule [110] is adopted for the calculation of integrals. The integral term in Eq. (4.2) can be computed as

$$\int_0^{t_f} \dddot{q}^T(t) R \dddot{q}(t) dt = \sum_{i=0}^{N} w_i \dddot{q}^T(K_i) R \dddot{q}(K_i) \tag{4.17}$$

where w_i is the quadrature weight at the ith interpolation knot as mentioned in [110], which is expressed as

$$w_i = \int_0^{t_f} \mathcal{L}_i(t) dt. \tag{4.18}$$

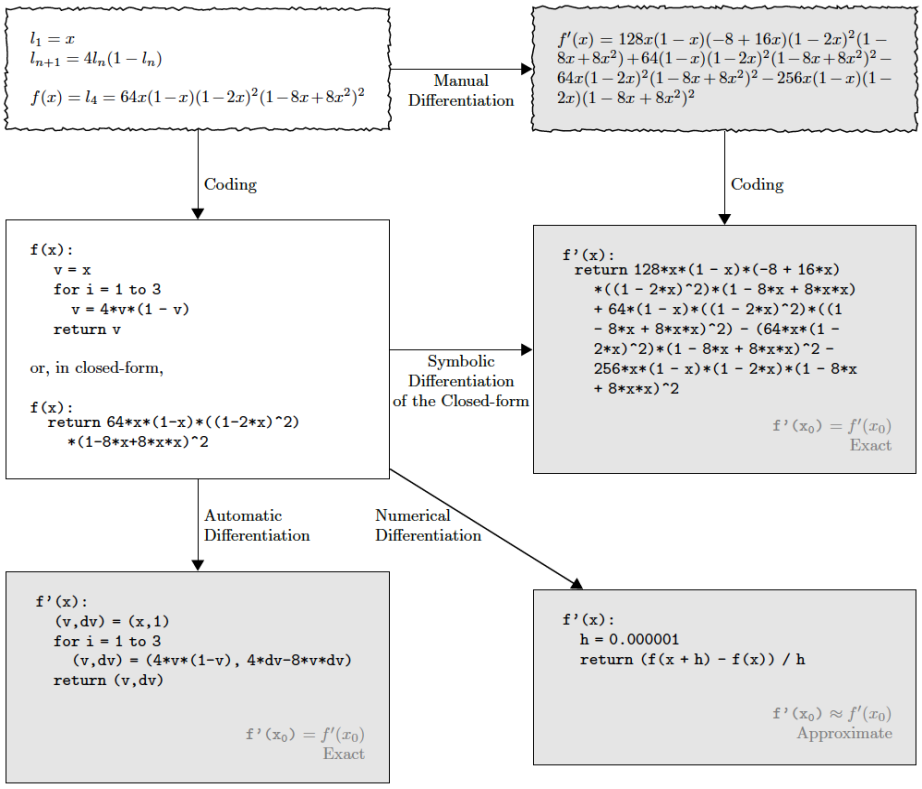


Figure 4.2: Principles of four differentiation approaches [111]: manual differentiation, numerical differentiation, symbolic differentiation and automatic differentiation.

Referring to $\xi \in [-1, 1]$, Eq. (4.18) can be rewritten as

$$w_i = \int_0^{t_f} \mathcal{L}_i(t)dt = \frac{t_f}{2}\int_{-1}^{1} \mathcal{L}_i'(\xi)d\xi \tag{4.19}$$

where

$$\mathcal{L}_i'(\xi) = \frac{(-1)^i}{\xi - \xi_i} \Bigg/ \left[\sum_{a=1}^{N-1} \frac{(-1)^a}{\xi - \xi_a} + \frac{1}{2}\left(\frac{1}{\xi - \xi_0} + \frac{(-1)^N}{\xi - \xi_N}\right)\right]. \tag{4.20}$$

The gradient-descent-based techniques are commonly utilized as the optimization solvers. In particular, interior point method has been successfully implemented for linear, conic and nonlinear optimization. The details of interior point method can be referred in [112].

Several approaches for the computation of derivatives have been applied in the field of

machine learning, as shown in Figure 4.2. Manual differentiation is prone to errors and time-consuming. Numerical differentiation may be highly inaccurate due to round-off errors and cause high computation consumption for the inputs with large dimensions [113]. Symbolic differentiation is able to avoid the problems of manual differentiation and numerical differentiation, but it can result in complex expressions and inefficient codes [113]. Numerical differentiation and symbolic differentiation also encounter difficulties for calculating high-order derivatives. To address the above problems, automatic differentiation has been developed and broadly applied for engineering design optimization. By incorporating derivative values and redefining the operator semantics to propagate derivatives according to the chain rule of differentiation calculus [111], it performs better than the other three differentiation methods. To apply interior point method and automatic differentiation, CasADi [114] is incorporated in MATLAB to solve the NLP in this work.

4.5 Vision-based Trajectory Generation of the 6-RSS Parallel Robot

In terms of the desired fiber path AB on Branch 2 in Figure 1.5, a time-jerk optimal trajectory can be planned for the AFP head from point A to point B subject to the dynamic and kinematic constraints of the serial robot. Under this circumstance, the AFP head path is deviated from the desired fiber path, and the direction of the roller in the AFP head cannot meet the demand of keeping perpendicular to the mandrel surface. To address this problem, the parallel robot needs to move to compensate the serial robot motion in order to satisfy the AFP geometric constraints.

The trajectory planning in this section aims to maintain the kinematic relationships between the roller in the AFP head and the Y-shape mandrel surface. In order to accomplish the AFP task, the desired trajectory of the start point A on the mandrel can

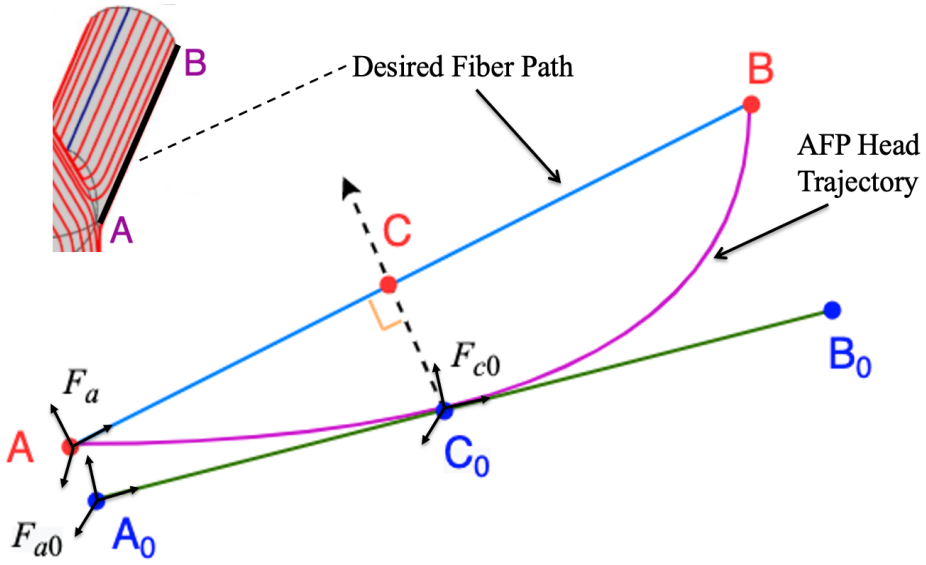


Figure 4.3: Trajectory generation approach for the start point A.

be defined. Thanks to C-Track 780, the pose of different frames w.r.t. F_s in Figure 1.4 can be measured. As a result, the desired trajectory of the parallel robot end-effector can be generated thereafter. The trajectory generation approach for the start point A is illustrated in Figure 4.3, where ACB denotes the desired fiber path, and AC_0B denotes the time-jerk optimal trajectory for the AFP head. Assuming there are m waypoints on AC_0B along the time slot $t = [t_1, t_2, ..., t_m]^T$, and the corresponding positions of F_r are $\zeta_j = [x_j^r, y_j^r, z_j^r]^T, j = 1, 2, ..., m$. With the origin located at point A, F_a is defined with the same orientations as F_r along the three axes at the initial time. The homogeneous transformation matrix between F_a and F_{ep} should remain constant during the AFP process, which will be referred as T_0 in the following parts. When the AFP head moves to point C_0 at time t_c, a frame F_{c0} can be defined the same as F_r in all the six dimensions. The tangent at point C_0 can be obtained in 3D space with the direction vector $P = [p_1, p_2, p_3]^T$ given as

$$p_i = \frac{\zeta_c(i) - \zeta_{c-1}(i)}{t_c - t_{c-1}}, i = 1, 2, 3. \tag{4.21}$$

The Y-shape mandrel is a rigid model, and the roller direction should be perpendicular

to the mandrel surface at time t_c, thus we can define $\overrightarrow{AB} \perp \overrightarrow{C_0C}$ and $\left|\overrightarrow{A_0C_0}\right| = \left|\overrightarrow{AC}\right|$, then the position of point A_0 in Figure 4.3 can be determined. At time t_c, the start point A needs to be located at the position of point A_0 such that the AFP geometric constraints can be satisfied. Therefore, the desired trajectory of point A within the time slot t can be generated.

In Figure 4.3, F_{a0} is defined with the origin located at point A_0, thus the translation $[x_{ac}^0, y_{ac}^0, z_{ac}^0]^T$ between the origins of F_{a0} and F_{c0} is known, and the homogeneous transformation matrix between the two frames can be computed as

$$ {}^{c0}_{a0}T = \begin{bmatrix} 1 & 0 & 0 & x_{ac}^0 \\ 0 & 1 & 0 & y_{ac}^0 \\ 0 & 0 & 1 & z_{ac}^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.22} $$

The desired homogeneous transformation matrix ${}^{s}_{ep}T$ between F_{ep} and F_s at time t_c can be calculated by

$$ T_0 = {}^{s}_{ep}T^{-1}\,{}^{s}_{c0}T\,{}^{c0}_{a0}T \tag{4.23} $$

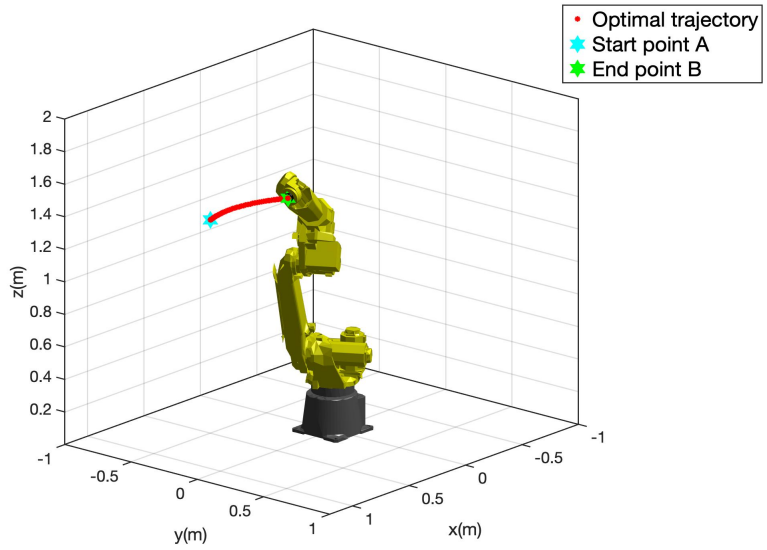
where ${}^{s}_{c0}T$ is the homogeneous transformation matrix between F_{c0} and F_s.

The desired homogeneous transformation matrix between F_{ep} and F_{bp} at time t_c can be obtained as

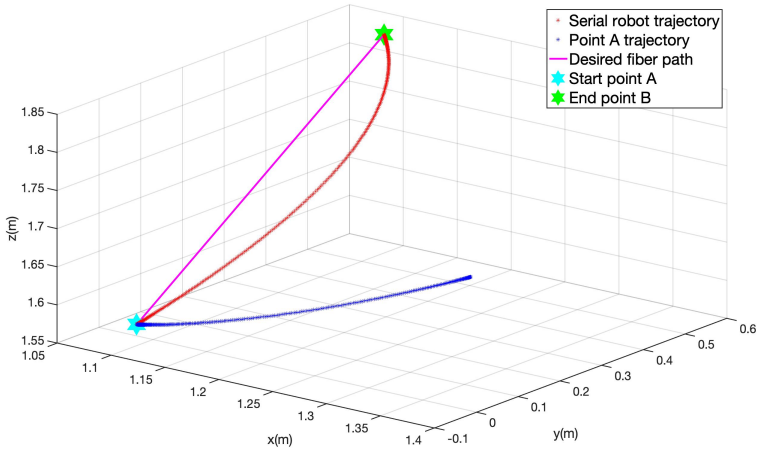
$$ {}^{bp}_{ep}T = {}^{s}_{bp}T^{-1}\,{}^{s}_{ep}T \tag{4.24} $$

where ${}^{s}_{bp}T$ is the homogeneous transformation matrix between F_{bp} and F_s. Therefore, the desired end-effector pose of the parallel robot at time t_c is determined. Based on such principle, the desired end-effector trajectory within the time slot t can be obtained, which will be utilized as the reference trajectory for the control of the parallel robot.

The desired end-effector motion of the parallel robot needs to satisfy the workspace



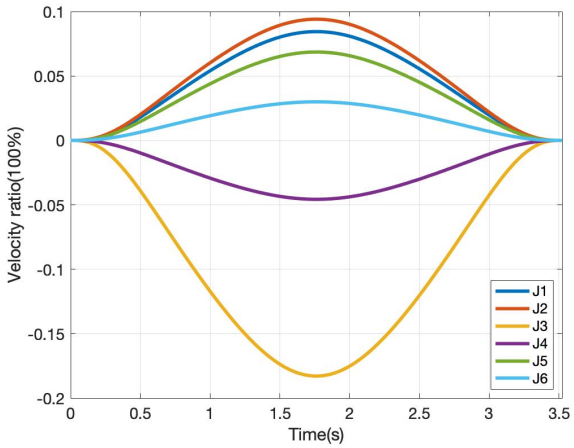
(a)



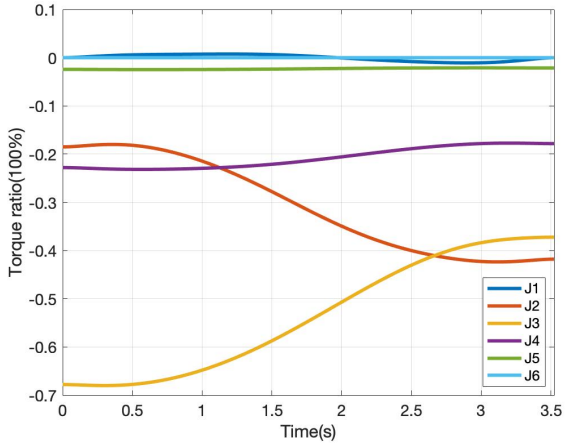
(b)

Figure 4.4: Simulation results of the leader-follower trajectory planning approach.

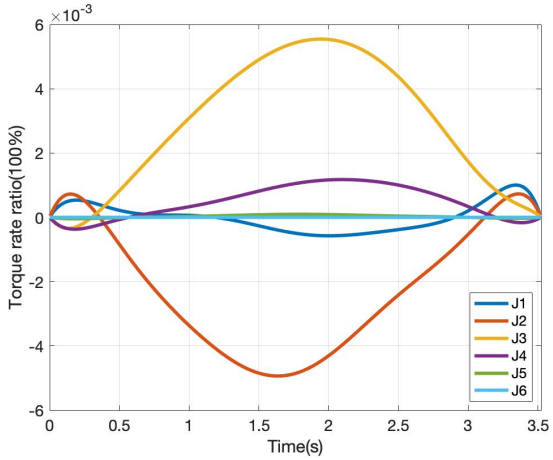
limit analyzed in [106]. The joint states of the parallel robot can be uniquely determined from the end-effector pose based on inverse kinematics. The desired joint states should be calculated to ensure that the actuator motion is within the singularity-free and bifurcation-free range $(-0.9948\,\mathrm{rad}, 0.9948\,\mathrm{rad})$ [107].



(a)



(b)



(c)

Figure 4.5: Scaled joint velocities, torques and torque rates in the simulation.

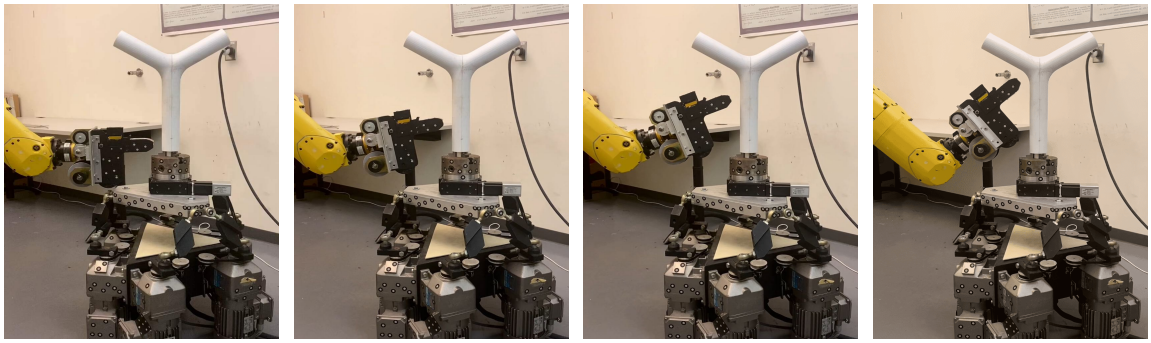


Figure 4.6: AFP head movements along Branch 1 and the intersection part between Branch 1 and Branch 2 at different moments in Test 1.

4.6 Simulation and Experimental Results

4.6.1 Simulation Tests

In the simulation, the time-jerk optimal trajectory planning method for the serial robot and the trajectory generation approach for the start point A are validated. The modeling of the AFP head is not considered, and F_r is defined the same as F_{es}. The initial and final joint states of the serial robot are designed as $[0, 0, 0, 0, 0, 0]^T$ and $[\pi/6, \pi/6, -\pi/3, -\pi/6, \pi/4, \pi/6]^T$, respectively, λ in Eq. (4.2) is given as 0.3, and the number of knots in pseudospectral method is defined as 10. The positions of the start point A and the end point B can be calculated using robot forward kinematics. The optimal trajectory is planned for the serial robot first, then the desired trajectory of the start point A is generated.

The trajectory planning results are shown in Figure 4.4. It is obvious that the planned trajectories for the serial robot and point A are smooth, which is crucial to finish the AFP task in real applications. The scaled joint velocities, torques and torque rates along the optimal trajectory are illustrated in Figure 4.5. It can be seen that they are all bounded within the actuator limits, thus the optimal trajectory planning method is implementable for the serial robot. The motion time is 3.5218 s, which can be adjusted by tuning λ. The smaller λ will lead to smoother trajectory but longer motion time.

4.6.2 Experimental Tests

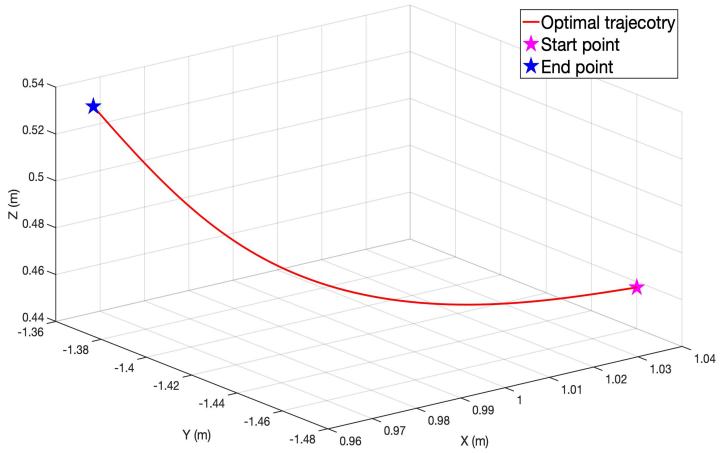
The experimental tests focus on validating the trajectory planning strategy for the cooperative robotic system without laying the real fiber on the mandrel surface. The geometric constraints that should be satisfied in the experiment include two aspects: (1) The roller direction should be perpendicular to the mandrel surface during the motion. (2) A small constant distance $d = 0.015$ m along the roller direction is defined and needs to be maintained between the roller and the mandrel surface. The trajectory planning strategy is tested with a continuous fiber path along Branch 1 and 2 on the Y-shape mandrel, as indicated in Figure 1.5.

The AFP head movements along Branch 1 and the intersection part of the two branches at different moments are illustrated in Figure 4.6, where the AFP head can successfully follow the desired fiber path. In order to pass the intersection part, the constant distance d is maintained but the roller direction is not strictly perpendicular to the mandrel surface along Branch 1. In this case, the desired fiber path can still be followed due to the fact that the roller is with a cylindrical shape. While for Branch 2, the roller direction will be set as perpendicular to the mandrel surface at the start point and the end point. The time-jerk optimal trajectory will be planned for the AFP head first, then the desired trajectory of the parallel robot will be generated. The comparisons will be made between the leader-follower trajectory planning strategy and the trajectory planning algorithm in [14].

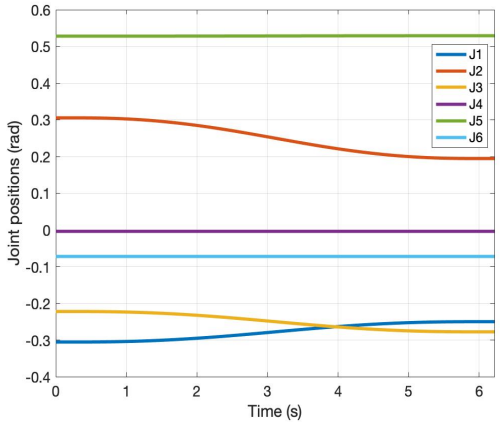
When the AFP head is at the start point and the end point, the initial joint state q^0 and the final joint state q^f of the serial robot can be obtained as

$$\begin{cases} q^0 = [-55^\circ, 55^\circ, -40^\circ, -0.7^\circ, 95^\circ, -13^\circ]^T, \\ q^f = [-45^\circ, 35^\circ, -50^\circ, -0.7^\circ, 95^\circ, -13^\circ]^T. \end{cases} \tag{4.25}$$

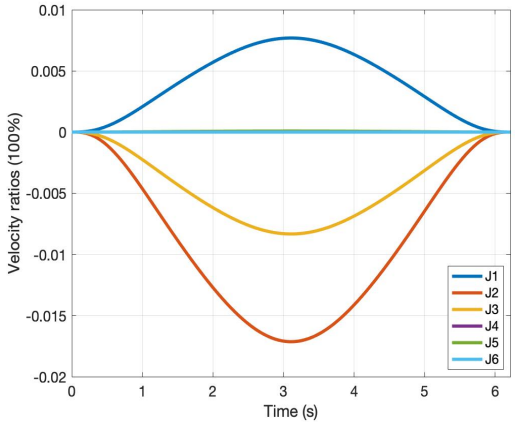
Test 1: The motion time regarding the optimal trajectory planning along Branch 2 can be adjusted by tuning λ in Eq. (4.2). Larger λ can lead to higher trajectory smoothness and



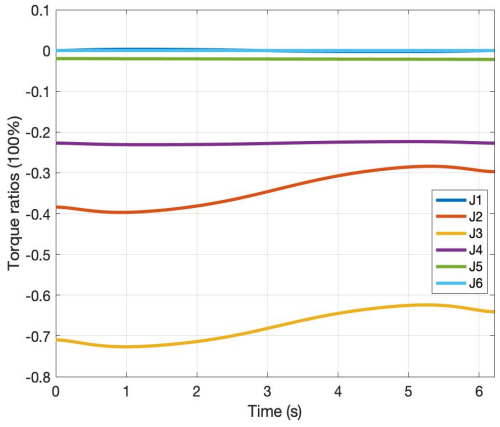
(a)



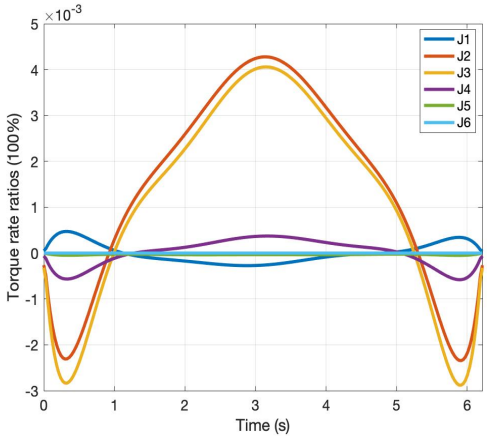
(b)



(c)

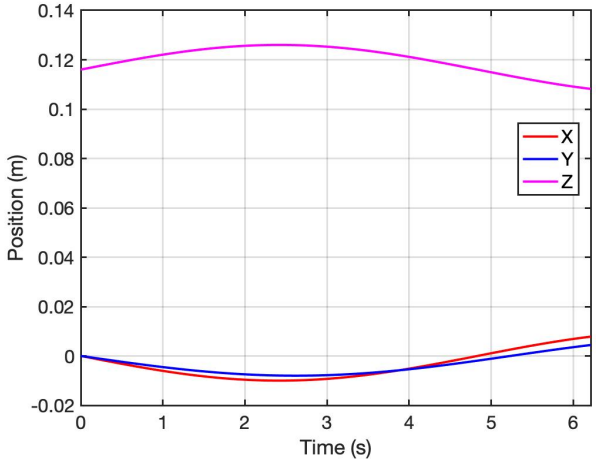


(d)

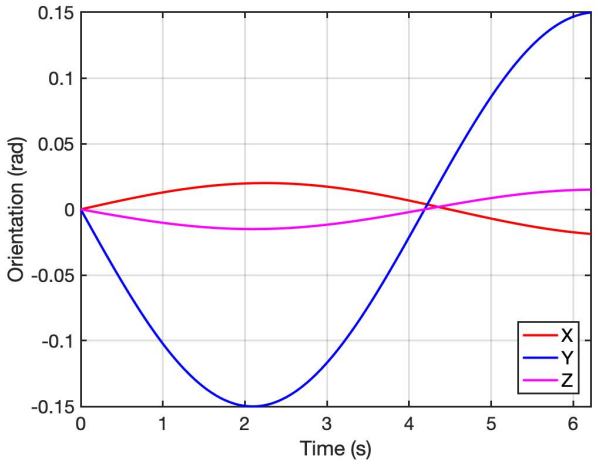


(e)

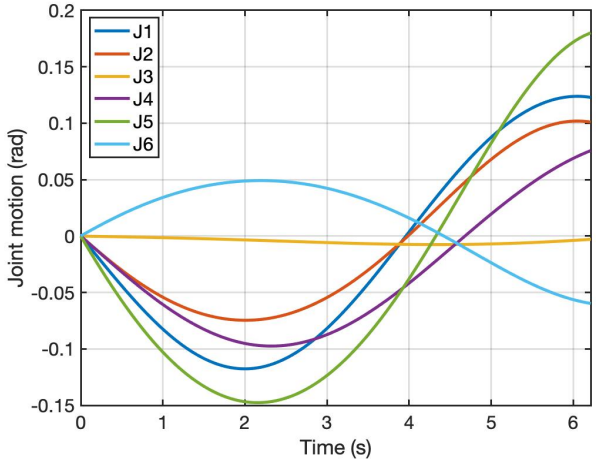
Figure 4.7: Time-jerk optimal trajectory planning results for the serial robot.



(a)



(b)



(c)

Figure 4.8: Desired end-effector pose and joint motion of the parallel robot.

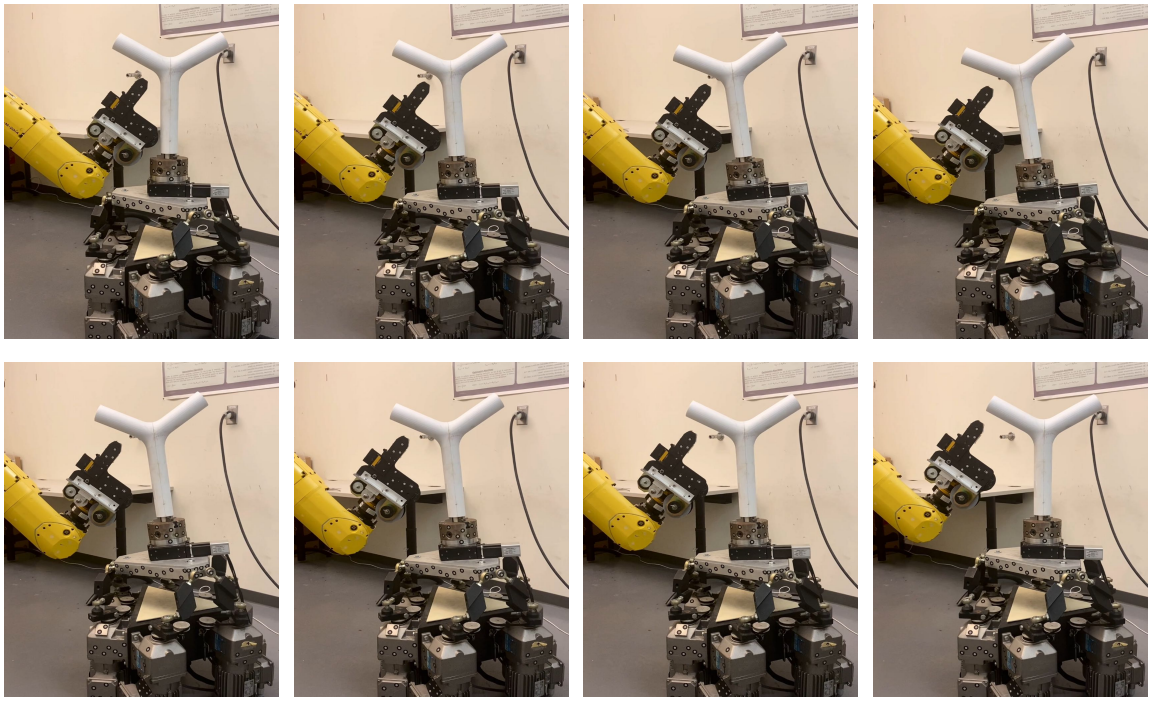
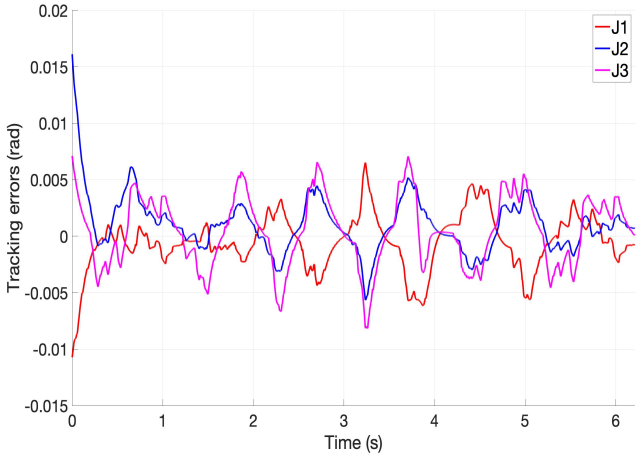
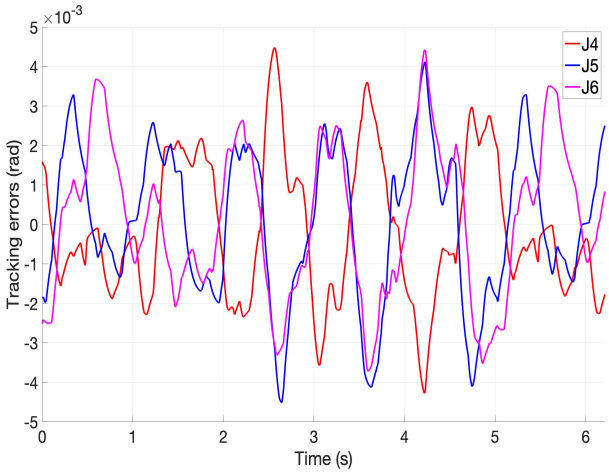


Figure 4.9: Cooperative motion of the two robots for the fiber placement on Branch 2 at different moments in Test 1.

longer motion time. In this experiment, λ is set as 0.08, and the number of interpolation knots in pseudospectral method is designed as $N+1 = 10$. The generated time-jerk optimal trajectory for the serial robot, the desired joint positions, and the scaled joint velocities, torques and torque rates are demonstrated in Figure 4.7. The motion time is 6.2128 s, and it is obvious that the joint velocities, torques and torque rates are all bounded within the robot actuator limits. According to the vision-based trajectory generation approach, the desired trajectory of the parallel robot, namely, the desired pose of F_{ep} w.r.t. F_{bp} can be generated, as shown in Figure 4.8 (a) and (b). Based on the inverse kinematics of the parallel robot, the desired joint motion can be calculated, as illustrated in Figure 4.8 (c). The desired end-effector motion is within the reachable workspace, and the joint motion is singularity-free and bifurcation-free.



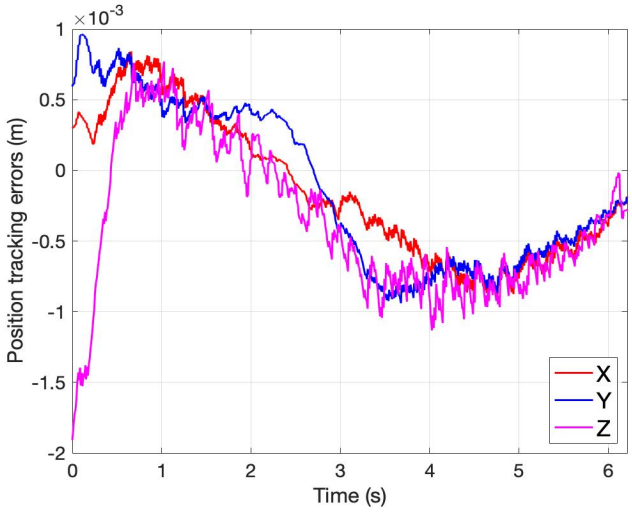
(a)



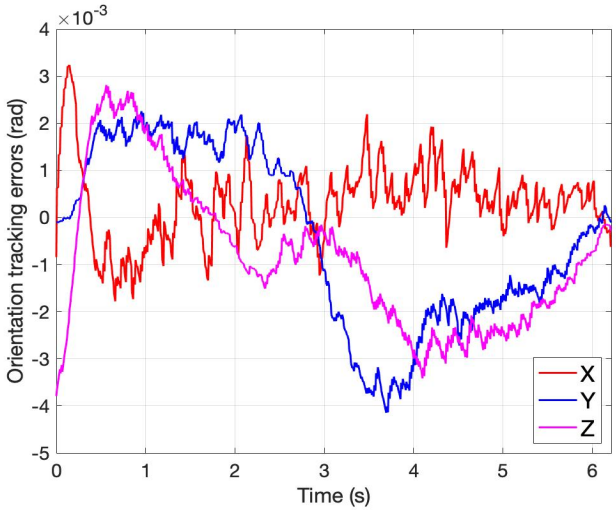
(b)

Figure 4.10: Joint position tracking errors of the serial robot in Test 1.

The cooperative motion of the two robots for the fiber placement on Branch 2 at different moments are demonstrated in Figure 4.9. The motion of the two robots are continuous and smooth without any interruptions, and the AFP geometric constraints can be maintained at the same time. The joint position tracking errors of the serial robot are presented in Figure 4.10. To further evaluate the tracking performance, the error indexes including mean absolute error (MAE) and root mean square error (RMSE) are presented in Table 4.1. The trajectory tracking can converge fast and small tracking errors can be



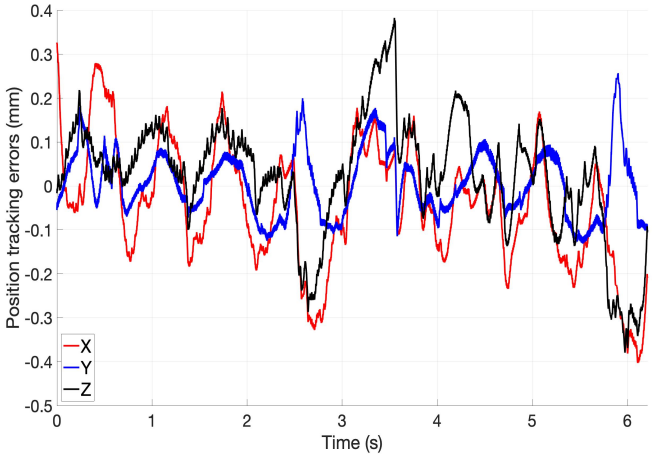
(a)



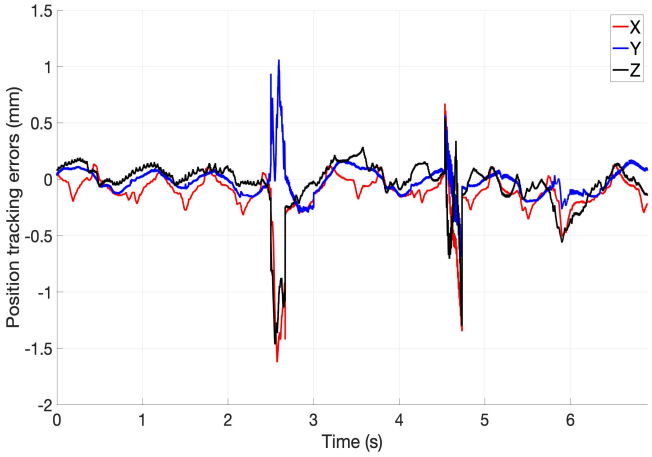
(b)

Figure 4.11: Pose tracking errors of the parallel robot in Test 1.

achieved for all the six joints. No actuator limits, joint singularities or joint limits have been encountered. The planned time-jerk optimal trajectory is implementable for the serial robot. The pose tracking errors of the parallel robot end-effector are shown in Figure 4.11, and the error indexes MAE and RMSE are given in Table 4.2. Good convergence speed and high tracking accuracy can be realized for all the six dimensions. A video of the leader-follower trajectory planning results in Test 1 can be found through the following



(a)



(b)

Figure 4.12: Position tracking errors of the serial robot end-effector in the two tests. (a) Test 1. (b) Test 2.

link https://www.youtube.com/watch?v=yOt-Kp1m0J0.

Test 2: As indicated in [14], while the AFP head is following the desired fiber path on Branch 2, joint 5 will reach to its limit 120°, which makes the robot stop at the limit point and interrupts the AFP process. In this test, when joint 5 reaches to 119°, the offset motion of the parallel robot is generated, and then the compensation motion of the serial robot is calculated. In this way, the limit point of joint 5 can be avoided and the serial robot can complete the rest of the trajectory. Since the trajectory planning of the serial robot is done

Table 4.1: Joint position tracking error indexes of the serial robot in Test 1

Position (10^{-3} rad)	**MAE**	**RMSE**
Joint 1	1.82	2.5
Joint 2	1.9	2.64
Joint 3	2.55	3.1
Joint 4	1.49	1.79
Joint 5	1.61	1.91
Joint 6	1.63	1.92

in Cartesian space, the end-effector speed is set as 30 mm/s. The total motion time in Test 2 is around 6.9 s.

The comparisons between the two tests are made by evaluating the position tracking performance of the serial robot end-effector. The tracking errors are shown in Figure 4.12, and the error indexes MAE and RMSE are given in Table 4.3. Good convergence speed can be achieved for the trajectory tracking of the serial robot in both tests, and smaller tracking errors can be obtained in Test 1. In Test 2, there are sharp increases and oscillations of the tracking errors during the time intervals 2.5 s to 2.65 s and 4.55 s to 4.75 s. This phenomenon is caused by the start and the end of the compensation motion of the serial robot to maintain the kinematic relationships to the parallel robot. Compared with the trajectory planning algorithm in [14], which generates the offset motion of the parallel robot and the compensation motion of the serial robot in real time, the advantages of the proposed trajectory planning strategy can be summarized as follows: (1) The trajectory of the serial robot is smoother without any sudden changes and higher tracking accuracy can be realized. (2) The trajectories of the serial robot and the parallel robot can be pre-planned, which guarantees better cooperation performance. The generated trajectories can

Table 4.2: Pose tracking error indexes of the parallel robot in Test 1

Pose	**MAE**	**RMSE**
X position (mm)	0.47	0.53
Y position (mm)	0.54	0.59
Z position (mm)	0.56	0.65
X orientation (10^{-3} rad)	0.67	0.88
Y orientation (10^{-3} rad)	1.68	2.06
Z orientation (10^{-3} rad)	1.54	1.79

Table 4.3: Position tracking error indexes of the serial robot end-effector in the two tests

Position (mm)	**Indexes**	**Test 1**	**Test 2**
X position	MAE	0.12	0.16
	RMSE	0.15	0.26
Y position	MAE	0.07	0.11
	RMSE	0.08	0.15
Z position	MAE	0.1	0.13
	RMSE	0.14	0.24

be used as the reference inputs for robot control in real applications. It provides flexibility for designing nonlinear controllers to improve the trajectory tracking performance.

4.7 Summary

This chapter introduces a novel trajectory planning strategy in a leader-follower formation for the 13-DOF cooperative AFP system. The objective is to finish the AFP task on the Y-shape mandrel along the desired 0° path without any interruptions. Taking into consideration the robot dynamic and kinematic constraints, a time-jerk optimal trajectory planning scheme is designed for the 6-DOF serial robot, which holds the AFP head and serves as the leader to place fiber. Pseudospectral method, interior point method, and automatic differentiation are employed for solving the trajectory optimization problem. In order to compensate the serial robot motion and satisfy the AFP geometric constraints, a vision-based trajectory generation approach is proposed for the 6-RSS parallel robot, which holds the mandrel using the 1-DOF rotary stage and works as the follower. Simulation and experimental results demonstrate the effectiveness and superiority of the proposed cooperative trajectory planning strategy.

Chapter 5

PBVS-based Adaptive Sliding Mode Control of the 6-RSS Parallel Robot

5.1 Problem Formulation

The trajectory tracking control of parallel robots is a challenging task due to their complicated kinematics and dynamics. The joint-space controller based on inverse kinematics has attracted researchers' attention. The joint positions of a parallel robot can be uniquely determined from the end-effector pose through inverse kinematics [115]. Nevertheless, the tracking accuracy of the joint-space controller can be decreased due to the inverse kinematics modeling errors. The Cartesian-space controller has also been popularly investigated, which can directly regulate the control of the end-effector pose to guarantee the desired performance. Unfortunately, the forward kinematics solution of a parallel robot is not unique, which means that the same joint positions may lead to various end-effector poses [116]. In this case, if the end-effector pose is estimated in an appropriate way, the calculation of forward kinematics can be avoided [117].

The dynamic behavior of a parallel robot is anisotropic and nonlinear due to various system uncertainties and dynamic coupling between the serial chains [118]. Consequently,

the pure kinematics-based controllers may not be able to meet the increasing demands of high accuracy and stability in real applications. To realize the desired performance of a parallel robot more efficiently, the dynamics of the robot should be considered in the controller design. In this chapter, the control of the 6-RSS parallel robot in the AFP system is studied. The kinematic parameter errors normally occur after the manufacturing of the robot, and the approximate dynamic parameters of the robot are provided by the manufacturer specifications. The uncertainties of the kinematic and dynamic parameters have negative impacts on the controller design and performance. By determining more accurate kinematic parameters, kinematic calibration can increase the positioning accuracy of the robot end-effector. Moreover, dynamic identification is a promising approach to determine the uncertain dynamic parameters with higher accuracy. The dynamic identification and kinematic calibration of the 6-RSS parallel robot have been done in [119].

Generally, the discrepancies between the nominal model and the actual robot can deteriorate the trajectory tracking accuracy and cause system oscillations. Sliding mode control techniques owning strong robustness to system uncertainties have been applied for parallel robots to eliminate such discrepancies. In [120], an improved adaptive fuzzy sliding mode controller for the position and force control of a Stewart robot is introduced. An integral sliding mode controller with a continuous twisting control algorithm is developed for a 6-DOF Stewart platform in [121]. In [122], a sliding mode control method with an improved equivalent-input-disturbance is designed for a Stewart platform to enhance the disturbance-rejection and tracking performance. Additionally, neural network has been extensively applied in the control of parallel robots. Equipped with strong self-learning and approximation capability, neural network is widely implemented for the auto-tuning of the Proportional-Integral-Derivative (PID) control gains. An RBF-neural-network-based PID controller with online tuning parameters is proposed for a Stewart platform in [123]. To address the issue of vibration isolation, an adaptive PID controller using single neuron is

developed for a hydraulically actuated Stewart robot in [124]. In [125], the parameters of a PID controller for a SCARA parallel robot with pneumatic artificial muscle actuator are tuned based on back-propagation (BP) neural network. In industry, parallel robots track various trajectories with different speeds, and the system frictions and disturbances are time-varying [126]. Under this circumstance, the online updates of the control gains in sliding mode controllers are necessary in order to achieve the desired performance. In [127–129], researchers incorporate neural networks into sliding mode controllers to achieve the auto-tuning of the control gains for serial robots, but the sufficient stability analyses of the controllers are not provided. Moreover, similar work for parallel robots has been mostly done regarding the computed torque controllers without involving the sliding mode control techniques, as introduced in [69, 130–132]. To this day, for the combination of sliding mode control techniques and neural networks in the field of parallel robots, the majority of the work employs neural networks to estimate the dynamic uncertainties in the control laws, as shown in [76, 133–135]. Furthermore, in terms of the visual servoing of parallel robots, not much work integrating sliding mode control has been published in the past decades.

5.2 Contributions

For the purpose of enhancing the trajectory tracking performance, this chapter develops a position-based visual servoing strategy using an adaptive sliding mode controller for the 6-RSS parallel robot. It can realize highly accurate trajectory tracking based on the real-time pose measurements from the photogrammetry sensor C-Track 780 in the eye-to-hand configuration. The detailed contributions are described as below:

(1) Compared with [85, 136], where only the kinematics of the robot is considered in PBVS, the proposed PBVS approach takes into account the full kinematics and dynamics of the parallel robot to obtain better tracking performance. The adaptive Kalman filter in

Chapter 3 is adopted to cope with uncertain noises in pose estimation.

(2) Sliding mode control technique is utilized to deal with dynamic uncertainties to guarantee system robustness. Due to the fact that parallel robots work with time-varying conditions in real applications, RBF neural network is employed for the auto-tuning of the control gains. From [69, 76, 127–135], it can be seen that this topic has been rarely investigated for parallel robots. Additionally, the dynamics of the parallel robot is modeled using the dynamic identification results in [119].

(3) Different from [124, 125, 127–132], where single neuron, multi-layer neural network and BP neural network are used for the auto-tuning of the control gains, RBF neural network is adopted in this project. By employing radial basis function as the activation function, RBF neural network works as the local approximator, which demonstrates stronger learning ability and higher approximation accuracy and speed with a simple structure [59].

(4) In [123], RBF neural network is utilized in PID controller, but the stability analysis of the controller is not provided. In terms of the adaptive sliding mode controllers for serial robots in [127–129], the stability analyses are not done or just partly presented. In this project, the detailed stability analysis of the proposed adaptive sliding mode controller is provided based on Lyapunov theorem.

(5) Experiments have been done to demonstrate the effectiveness of the PBVS scheme. Simulation and experimental results illustrate the superiority of the designed controller over the sliding mode controller, and the adaptive sliding mode controllers using BP neural network and single neuron, respectively.

In the rest of this chapter, a brief description of the system model and the PBVS scheme is given, and the adaptive sliding mode control strategy is discussed. Simulation and experimental results are presented, and a summarized conclusion is provided.

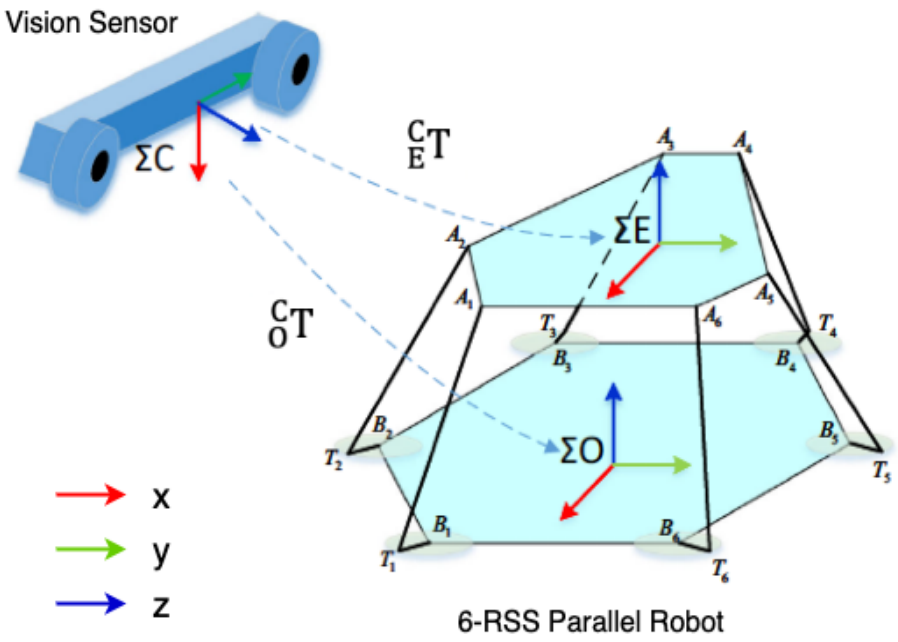


Figure 5.1: Position-based visual servoing system.

5.3 System Description

The schematic of the PBVS system, including a 6-RSS parallel robot and a dual camera in the eye-to-hand configuration, is illustrated in Figure 5.1. The vision sensor frame with the origin at the center of the camera is defined as $\sum C$. The end-effector frame and the base frame of the parallel robot are denoted as $\sum E$ and $\sum O$, respectively, whose origins are chosen as the symmetrical centers of the platforms. The dual camera is employed to measure the end-effector pose $X = [\psi^T, \phi^T]^T = [x, y, z, \alpha, \beta, \gamma]^T \in R^6$ w.r.t. $\sum O$ in real time, where $\psi = [x, y, z]^T$ is the position variable, and $\phi = [\alpha, \beta, \gamma]^T$ is the Euler angle. The angular velocity $\omega \in R^3$ and velocity $v \in R^6$ can be calculated as

$$\omega = \begin{bmatrix} 1 & 0 & s\beta \\ 0 & c\alpha & -c\beta s\alpha \\ 0 & s\alpha & c\beta c\alpha \end{bmatrix} \dot{\phi} = J_e \dot{\phi} \tag{5.1}$$

$$v = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & J_e \end{bmatrix} \dot{X} = J_s \dot{X}. \tag{5.2}$$

where $s\alpha = \sin(\alpha)$, $c\alpha = \cos(\alpha)$, and J_e and J_s are Jacobian matrices.

The dynamic model of the parallel robot is represented as [10]

$$M(X)\ddot{X} + C(X,\dot{X})\dot{X} + G(X) + \tau_f = \tau \tag{5.3}$$

where $\dot{X} \in R^6$ and $\ddot{X} \in R^6$ represent end-effector velocity and acceleration, respectively. $M(X) \in R^{6\times6}$, $C(X,\dot{X}) \in R^{6\times6}$, $G(X) \in R^6$, $\tau_f \in R^6$ and $\tau \in R^6$ denote inertia matrix, Coriolis and centrifugal matrix, gravitational term, frictional term, and control torque, respectively. The detailed dynamic and kinematic models of the parallel robot can be found in [119].

The following properties of the dynamic equation will be employed for the controller development in this work [10].

Property 1: $M(X)$ is a symmetric and uniformly positive definite matrix.

Property 2: $\dot{M}(X) - 2C(X,\dot{X})$ is a skew-symmetric matrix such that $W^T(\dot{M}(X) - 2C(X,\dot{X}))W = 0$ for any $W \in R^6$.

5.4 Position-based Visual Servoing Scheme

The position-based visual servoing scheme for the 6-RSS parallel robot in Cartesian space is presented in Figure 5.2. It mainly includes four parts: 6-RSS parallel robot with the built-in controller, adaptive sliding mode control strategy, torque to position transformer, and vision-based pose estimation method.

The PBVS control problem in Cartesian space can be stated as: given a desired end-effector trajectory $X_d(t)$, an adaptive sliding mode controller is developed to make the real

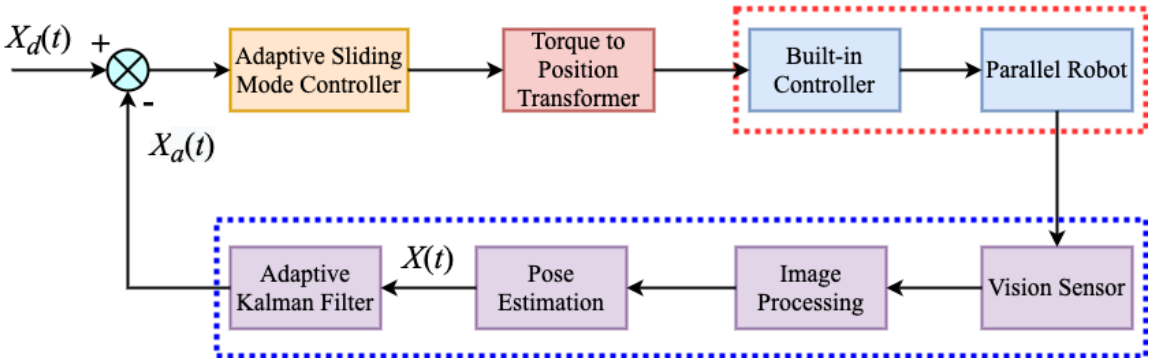


Figure 5.2: Position-based visual servoing scheme.

end-effector pose $X_a(t)$ track the desired trajectory. For industrial robots, only the built-in controller can get access to the robot parameters and generate the motion commands. The torque to position transformer for the 6-RSS prallel robot has been designed in [119]. It can covert the torque signals sent by the adaptive sliding mode controller to the position signals the robot can use. Through image processing, the C-Track 780 can send out the measured end-effector pose of the parallel robot in a continuous time sequence. Then, the adaptive Kalman filter is adopted to estimate the uncertain noises in visual measurements. The filtered end-effector pose serves as the feedback signal to the adaptive sliding mode controller. The pose estimation method using C-Track 780 and adaptive Kalman filter have been introduced in Chapter 3.

5.5 Adaptive Sliding Mode Control

The structure of the RBF-neural-network-based adaptive sliding mode control strategy is shown in Figure 5.3. The details are introduced as follows.

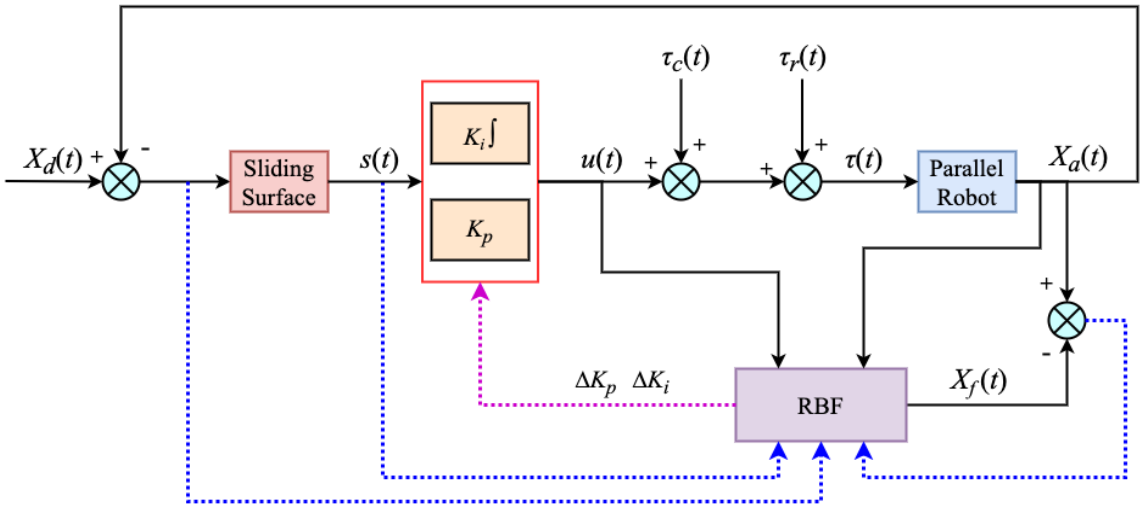


Figure 5.3: Adaptive sliding mode control strategy.

5.5.1 Sliding Mode Controller

The sliding mode controller is designed to cope with dynamic uncertainties in the robotic system. The pose and velocity tracking errors of the end-effector are defined as

$$\begin{aligned} e &= X_d - X_a \\ \dot{e} &= \dot{X}_d - \dot{X}_a \end{aligned} \tag{5.4}$$

where X_a and $\dot{X}_a$ are the filtered pose and velocity by AKF, respectively, and X_d and $\dot{X}_d$ denote the desired pose and velocity, respectively.

The sliding surface is given as

$$s = \dot{e} + \gamma e \tag{5.5}$$

where $\gamma \in R^{6\times 6}$ is a positive diagonal matrix.

The sliding mode controller is developed as

$$\tau = \tau_c + u + \tau_r \tag{5.6}$$

where

$$\tau_c = M_0\left(X_a\right)\ddot{X}_r + C_0(X_a, \dot{X}_a)\dot{X}_r + G_0\left(X_a\right) + \tau_{f0} \tag{5.7}$$

$$u = K_p s + K_i \int_0^t s\, dt \tag{5.8}$$

$$\tau_r = K_r \mathrm{sat}\left(\frac{s}{\varepsilon}\right) \tag{5.9}$$

$\dot{X}_r = s + \dot{X}_a$, and $\ddot{X}_r = \ddot{X}_d + \gamma\dot{e}$. $M_0\left(X_a\right)$, $C_0(X_a, \dot{X}_a)$, $G_0\left(X_a\right)$, and τ_{f0} are the estimated dynamic parameters. $K_p \in R^{6\times6}$, $K_i \in R^{6\times6}$ and $K_r \in R^{6\times6}$ are positive diagonal matrices. $\mathrm{sat}(\cdot)$ is the saturation function to reduce the chattering problems, given as

$$\mathrm{sat}(\frac{s_i}{\varepsilon}) = \begin{cases} \dfrac{s_i}{\varepsilon}, & |s_i| \le \varepsilon \\ \mathrm{sgn}(s_i), & |s_i| > \varepsilon \end{cases} \tag{5.10}$$

where $i = 1, 2, ..., 6$, and ε is a small positive constant. The dynamic uncertainties are defined as

$$\begin{aligned} \Delta M =& M(X_a) - M_0(X_a) \\ \Delta C =& C(X_a, \dot{X}_a) - C_0(X_a, \dot{X}_a) \\ \Delta G =& G(X_a) - G_0(X_a) \\ \Delta\tau_f =& \tau_f - \tau_{f0}. \end{aligned} \tag{5.11}$$

5.5.2 Auto-tuning of the Control Gains Based on RBF Neural Network

In this work, RBF neural network is adopted due to its strong self-learning capability and high approximation accuracy. The designed RBF neural network structure for the parallel robot is demonstrated in Figure 5.4. It is composed of six single neural networks. There are three layers in each neural network structure: input layer, hidden layer and output layer, and there are six nodes in each hidden layer.

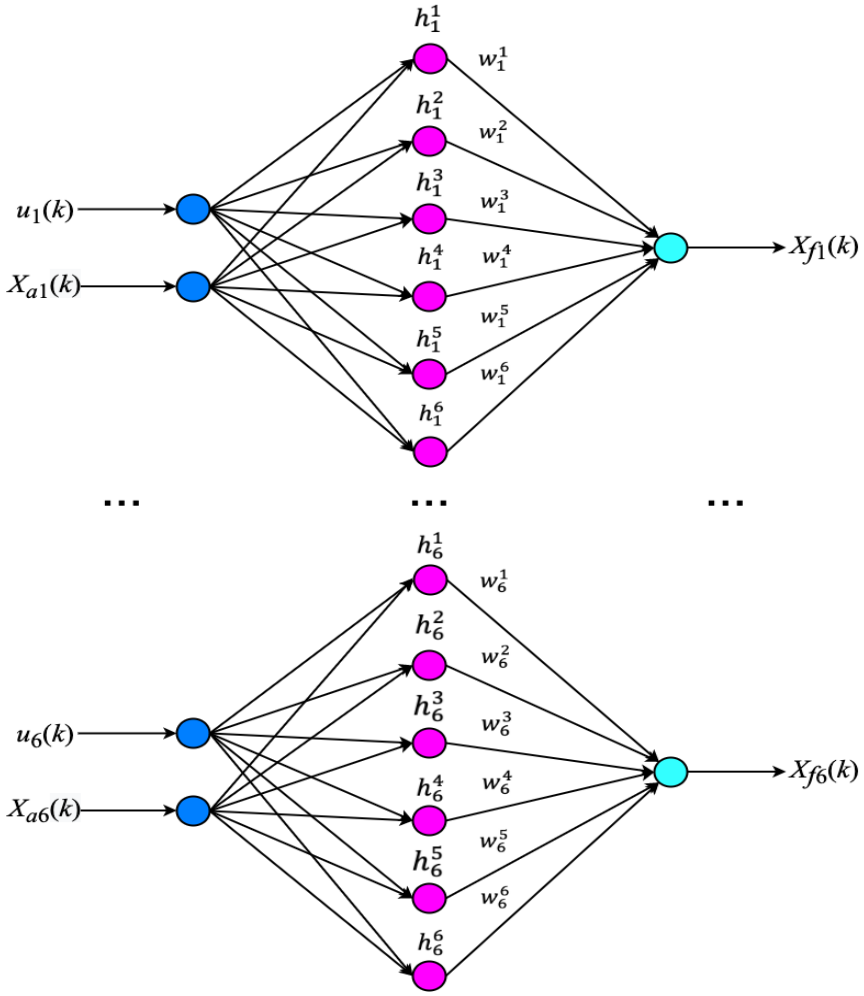


Figure 5.4: RBF neural network structure.

The activation function of the jth node in the hidden layer of the nth neural network is designed as Gaussian function

$$h_n^j(x) = \exp\left(\frac{-\|x_n - c_n^j\|^2}{2b_n^{j\,2}}\right), n, j = 1, 2, ..., 6 \tag{5.12}$$

where $x_n = [u_n(k), X_{an}(k)]^T$ is the input, b_n^j and c_n^j denote the width and the center vector, respectively, and k is the sampling step. The neural network output $X_{fn}(k)$ is computed as

$$X_{fn}(k) = \sum_{j=1}^{6} w_n^j h_n^j \tag{5.13}$$

where w_n^j is the weight linking the jth node in the hidden layer to the node in the output layer.

The RBF-neural-network-based auto-tuning approach includes two steps: identification and auto-tuning. In the first step, RBF neural network identifies the robot dynamics, and the neural network parameters are tuned. In the second step, the auto-tuning of K_p and K_i in Eq. (5.8) can be realized.

Regarding the dynamic identification, the index function is given as

$$J_n(k) = \frac{1}{2}(X_{an}(k) - X_{fn}(k))^2. \tag{5.14}$$

The gradient descent method is adopted to realize the automatic tuning of the parameters including weight, width and center vector. The update process is formulated as

$$\begin{aligned} w_n^j(k) &= w_n^j(k-1) + \Delta w_n^j + \alpha(w_n^j(k-1) - w_n^j(k-2)) \\ b_n^j(k) &= b_n^j(k-1) + \Delta b_n^j + \alpha(b_n^j(k-1) - b_n^j(k-2)) \\ c_n^{ji}(k) &= c_n^{ji}(k-1) + \Delta c_n^{ji} + \alpha(c_n^{ji}(k-1) - c_n^{ji}(k-2)) \end{aligned} \tag{5.15}$$

$$\begin{aligned} \Delta w_n^j &= \mu(X_{an}(k) - X_{fn}(k))h_n^j \\ \Delta b_n^j &= \mu(X_{an}(k) - X_{fn}(k))w_n^j h_n^j \|x_n - c_n^j\|^2 / {b_n^j}^3 \\ \Delta c_n^{ji} &= \mu\left(X_{an}(k) - X_{fn}(k)\right) w_n^j h_n^j (x_n^i - c_n^{ji}) / {b_n^j}^2 \end{aligned} \tag{5.16}$$

where α is the momentum coefficient, μ is the learning rate, and $i = 1, 2$. The Jacobian information representing the sensitivity of the system output to the change of the control input is calculated as

$$\frac{\partial X_{an}(k)}{\partial \Delta u_n(k)} \approx \frac{\partial X_{fn}(k)}{\partial \Delta u_n(k)} = \sum_{j=1}^{6} w_n^j h_n^j \frac{c_n^{j1} - \Delta u_n(k)}{{b_n^j}^2}. \tag{5.17}$$

For the auto-tuning process, the index function is defined as

$$E_n(k) = \frac{1}{2}(X_{dn}(k) - X_{an}(k))^2. \tag{5.18}$$

Based on the gradient descent method, the control gains can be tuned using

$$\begin{aligned} K_{pn}(k) &= K_{pn}(k-1) + \Delta K_{pn} \\ K_{in}(k) &= K_{in}(k-1) + \Delta K_{in} \end{aligned} \tag{5.19}$$

$$\Delta K_{pn} = -\eta \frac{\partial E_n(k)}{\partial K_{pn}(k)}, \Delta K_{in} = -\eta \frac{\partial E_n(k)}{\partial K_{in}(k)} \tag{5.20}$$

where η is the learning rate, and

$$\begin{aligned} \frac{\partial E_n(k)}{\partial K_{pn}(k)} &= \frac{\partial E_n(k)}{\partial X_{an}(k)} \frac{\partial X_{an}(k)}{\partial \Delta u_n(k)} \frac{\partial \Delta u_n(k)}{\partial K_{pn}(k)} \\ \frac{\partial E_n(k)}{\partial K_{in}(k)} &= \frac{\partial E_n(k)}{\partial X_{an}(k)} \frac{\partial X_{an}(k)}{\partial \Delta u_n(k)} \frac{\partial \Delta u_n(k)}{\partial K_{in}(k)} \end{aligned} \tag{5.21}$$

$$\begin{aligned} &\frac{\partial E_n(k)}{\partial X_{an}(k)} = -e_n(k), \frac{\partial \Delta u_n(k)}{\partial K_{in}(k)} = s_n(k), \\ &\frac{\partial \Delta u_n(k)}{\partial K_{pn}(k)} = s_n(k) - s_n(k-1). \end{aligned} \tag{5.22}$$

5.5.3 Stability Analysis

Based on Lyapunov theorem, the stability analysis of the adaptive sliding mode controller is conducted.

Lemma 1: Considering a nonlinear system with the differential equation $\dot{x} = f(x)$, and suppose $V(x)$ is a continuous function defined on an open neighborhood of the origin. If $V(x)$ is positive definite, and $\dot{V}(x) + c(V(x))^\beta \leq 0$, where $c > 0$ and $0 < \beta < 1$ are real numbers. The convergence time $T(x)$ of the control system starting from the initial state

$x(0)$ satisfies [137]

$$T(x(0)) \leq \frac{1}{c(1-\beta)} V(x(0))^{1-\beta}. \tag{5.23}$$

Theorem 1: For the closed-loop system in Figure 5.3 under the control of the sliding mode controller, if $|s_m| > \varepsilon, m = 1, 2, ..., 6$ is satisfied in Eq. (5.10), and then $K_r = \mathrm{diag}([K_{r1}, K_{r2}, ..., K_{r6}])$ in Eq. (5.9) can satisfy

$$K_{rm} \geq \sigma + |u_{pm}| + |u_{im}| + |\Delta H_m| \tag{5.24}$$

where σ is a small positive constant, $u_p = K_p s$, $u_i = K_i \int_0^t s \ dt$, $\Delta H = [\Delta H_1, \Delta H_2, ..., \Delta H_6]^T$, and $\Delta H = \Delta M \ddot{X}_r + \Delta C \dot{X}_r + \Delta G + \Delta \tau_f$, then the trajectory tracking convergence can be achieved.

Proof: The Lyapunov function of the sliding mode controller is defined as

$$V = \frac{1}{2} s^T M s. \tag{5.25}$$

By differentiating V w.r.t. time, we can get

$$\dot{V} = s^T \left(M\dot{s} + \frac{1}{2} \dot{M} s \right) = s^T \left(M\dot{s} + Cs \right). \tag{5.26}$$

Based on the designed controller, Eq. (5.3) can be formulated as

$$\begin{aligned} \tau &= M(\ddot{X}_r - \dot{s}) + C(\dot{X}_r - s) + G + \tau_f \\ &= M_0 \ddot{X}_r + C_0 \dot{X}_r + G_0 + \tau_{f0} + \Delta H - M\dot{s} - Cs. \end{aligned} \tag{5.27}$$

According to the kinematic calibration and dynamic identification results of the 6-RSS parallel robot in [119], it guarantees that ΔH is bounded as

$$|\Delta H_m| \leq H_m. \tag{5.28}$$

From Eqs. (5.7)-(5.10) and (5.27), one has

$$M\dot{s} + Cs = -K_p s - K_i \int_0^t s\, dt - K_r \mathrm{sgn}\,(s) + \Delta H. \tag{5.29}$$

Substituting Eq. (5.29) into Eq. (5.26) yields

$$\dot{V} = -s^T \left(K_p s + K_i \int_0^t s\, dt + K_r \mathrm{sgn}\,(s) - \Delta H \right). \tag{5.30}$$

If K_r satisfies Eq. (5.24), we can get

$$\dot{V} \leq -\sum_{m=1}^{6} \sigma \left| s_m \right| \leq -\sigma (s^T s)^{\frac{1}{2}} \leq 0. \tag{5.31}$$

As a result, $|s_m|$ will keep decreasing. According to *Lemma 1*, the boundary layer $|s_m| \leq \varepsilon$ can be reached within a finite time, and then s_m will remain inside. When $|s_m| = \varepsilon$, it can be obtained that $e_m = e_m(0)e^{-\gamma t} \pm \frac{\varepsilon}{\gamma}$, thus $|e_m| \to \left| \frac{\varepsilon}{\gamma} \right|$ and $\dot{e}_m \to 0$ with $t \to \infty$. Therefore, only the uniformly ultimately bounded stability rather than the asymptotic stability can be achieved.

Theorem 2: The stability analyses of an adaptive PID neural network controller and a PID-like neural network nonlinear adaptive controller have been done in [138] and [139], respectively. For the RBF-neural-network-based auto-tuning approach in this chapter, if Eqs. (5.32)-(5.35) can be satisfied, then the stability of the controller can be guaranteed.

(1) If $\sum_{n=1}^{6} (e_n^*(k-1)\Delta w_n^j(k-1)/h_n^j) < 0$, we have

$$\frac{\mu}{\alpha + 1} \leq \frac{1}{2} \frac{\sum_{n=1}^{6} \left(\frac{\Delta w_n^j(k-1)}{h_n^j} \right)^2}{\sum_{n=1}^{6} \left(\frac{e_n^*(k-1)\Delta w_n^j(k-1)}{h_n^j} \right)}. \tag{5.32}$$

If $\sum_{n=1}^{6}(e_n^*(k-1)\Delta w_n^j(k-1)/h_n^j) > 0$, we have

$$\frac{\mu}{\alpha+1} \geq \frac{1}{2}\frac{\sum_{n=1}^{6}\left(\frac{\Delta w_n^j(k-1)}{h_n^j}\right)^2}{\sum_{n=1}^{6}\left(\frac{e_n^*(k-1)\Delta w_n^j(k-1)}{h_n^j}\right)}. \tag{5.33}$$

(2) If $\sum_{n=1}^{6}(\Delta K_{pn}(k)q(k)s_n(k-1)/e_n(k)) < 0$, we have

$$\eta \geq -\frac{1}{2}\frac{\sum_{n=1}^{6}\left(\frac{\Delta K_{pn}(k)q(k)}{e_n(k)}\right)^2}{\sum_{n=1}^{6}\left(\frac{\Delta K_{pn}(k)q(k)s_n(k-1)}{e_n(k)}\right)}. \tag{5.34}$$

If $\sum_{n=1}^{6}(\Delta K_{pn}(k)q(k)s_n(k-1)/e_n(k)) > 0$, we have

$$\eta \leq -\frac{1}{2}\frac{\sum_{n=1}^{6}\left(\frac{\Delta K_{pn}(k)q(k)}{e_n(k)}\right)^2}{\sum_{n=1}^{6}\left(\frac{\Delta K_{pn}(k)q(k)s_n(k-1)}{e_n(k)}\right)}. \tag{5.35}$$

Proof: The Lyapunov function is given as

$$V(k) = V_1(k) + V_2(k). \tag{5.36}$$

$V_1(k)$ and $V_2(k)$ are defined as follows:

$$\begin{aligned} V_1(k) &= \frac{1}{2}\sum_{n=1}^{6} e_n^{*2}(k) \\ V_2(k) &= \frac{1}{2}\sum_{n=1}^{6} s_n^2(k) \end{aligned} \tag{5.37}$$

where $e_n^*(k) = X_{an}(k) - X_{fn}(k)$.

(1) The change of $V_1(k)$ is

$$\begin{aligned}\Delta V_1(k) &= V_1(k) - V_1(k-1) \\ &= \frac{1}{2}\sum_{n=1}^{6}\left(e_n^*(k) + e_n^*(k-1)\right)\left(e_n^*(k) - e_n^*(k-1)\right) \\ &= \sum_{n=1}^{6}\left(e_n^*(k-1)\Delta e_n^*(k) + \frac{1}{2}\Delta {e_n^*}^2(k)\right).\end{aligned} \tag{5.38}$$

From Eqs. (5.15) and (5.16), we can get

$$e_n^*(k) = \frac{1}{\mu h_n^j}((w_n^j(k) - w_n^j(k-1)) - \alpha(w_n^j(k-1) - w_n^j(k-2))). \tag{5.39}$$

Therefore

$$\Delta e_n^*(k) = \frac{\partial e_n^*(k)}{\partial w_n^j(k-1)}\Delta w_n^j(k-1) = -\frac{\alpha+1}{\mu h_n^j}\Delta w_n^j(k-1). \tag{5.40}$$

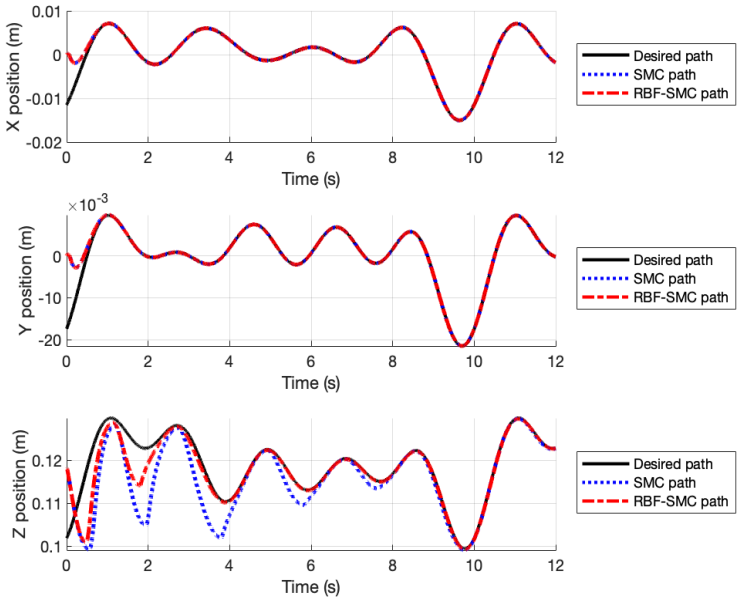
Then Eq. (5.38) can be formulated as

$$\Delta V_1(k) = -\frac{\alpha+1}{\mu}\sum_{n=1}^{6}\left(\frac{e_n^*(k-1)\Delta w_n^j(k-1)}{h_n^j}\right) + \frac{(\alpha+1)^2}{2\mu^2}\sum_{n=1}^{6}\left(\frac{\Delta w_n^j(k-1)}{h_n^j}\right)^2. \tag{5.41}$$

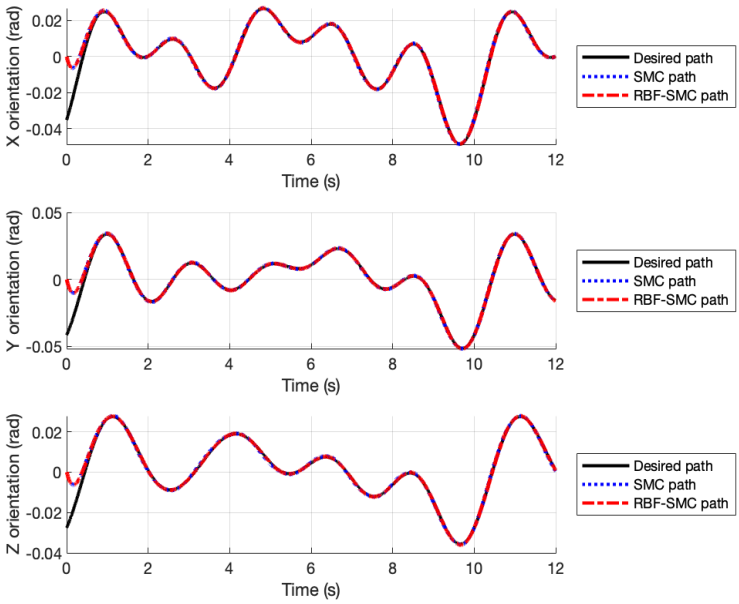
If μ and α can satisfy Eqs. (5.32) and (5.33), then $\Delta V_1(k) \leq 0$.

(2) The time difference of $V_2(k)$ is

$$\begin{aligned}\Delta V_2(k) &= V_2(k) - V_2(k-1) \\ &= \frac{1}{2}\sum_{n=1}^{6}(s_n(k) + s_n(k-1))(s_n(k) - s_n(k-1)) \\ &= \sum_{n=1}^{6}\left(s_n(k-1)\Delta s_n(k) + \frac{1}{2}\Delta s_n^2(k)\right).\end{aligned} \tag{5.42}$$



(a)



(b)

Figure 5.5: Trajectory tracking performance of the adaptive sliding mode controller with RBF-neural-network-based tuning method and the sliding mode controller with constant control gains.

In view of Eqs. (5.17), and (5.20)-(5.22), we can get

$$\begin{aligned}\Delta s_n(k) &= -\Delta K_{pn}(k) \Big/ \left(\eta \frac{\partial E_n(k)}{\partial X_{an}(k)} \frac{\partial X_{an}(k)}{\partial \Delta u_n(k)}\right) \\ &= \frac{\Delta K_{pn}(k)}{\eta e_n(k)} \sum_{j=1}^{6} \frac{{b_n^j}^2}{w_n^j h_n^j (c_n^{j1} - \Delta u_n(k))}.\end{aligned} \tag{5.43}$$

Let

$$q(k) = \sum_{j=1}^{6} \frac{{b_n^j}^2}{w_n^j h_n^j (c_n^{j1} - \Delta u_n(k))}. \tag{5.44}$$

We have

$$\Delta s_n(k) = \frac{\Delta K_{pn}(k) q(k)}{\eta e_n(k)}. \tag{5.45}$$

Then Eq. (5.42) can be rewritten as

$$\Delta V_2(k) = \frac{1}{\eta} \sum_{n=1}^{6} \frac{\Delta K_{pn}(k) q(k) s_n(k-1)}{e_n(k)} + \frac{1}{2\eta^2} \sum_{n=1}^{6} \left(\frac{\Delta K_{pn}(k) q(k)}{e_n(k)}\right)^2. \tag{5.46}$$

If η can satisfy Eqs. (5.34) and (5.35), then $\Delta V_2(k) \leq 0$.

Since $\Delta V_1(k) \leq 0$ and $\Delta V_2(k) \leq 0$, then $\Delta V(k) \leq 0$ can be satisfied for any sampling step k, thus the control system is stable.

5.6 Simulation and Experimental Results

5.6.1 Simulation Tests

In the simulation, the proposed adaptive sliding mode control approach is compared with the sliding mode controller with constant control gains. The desired end-effector trajectory is defined as the Finite Fourier series-based trajectory with varying frequency,

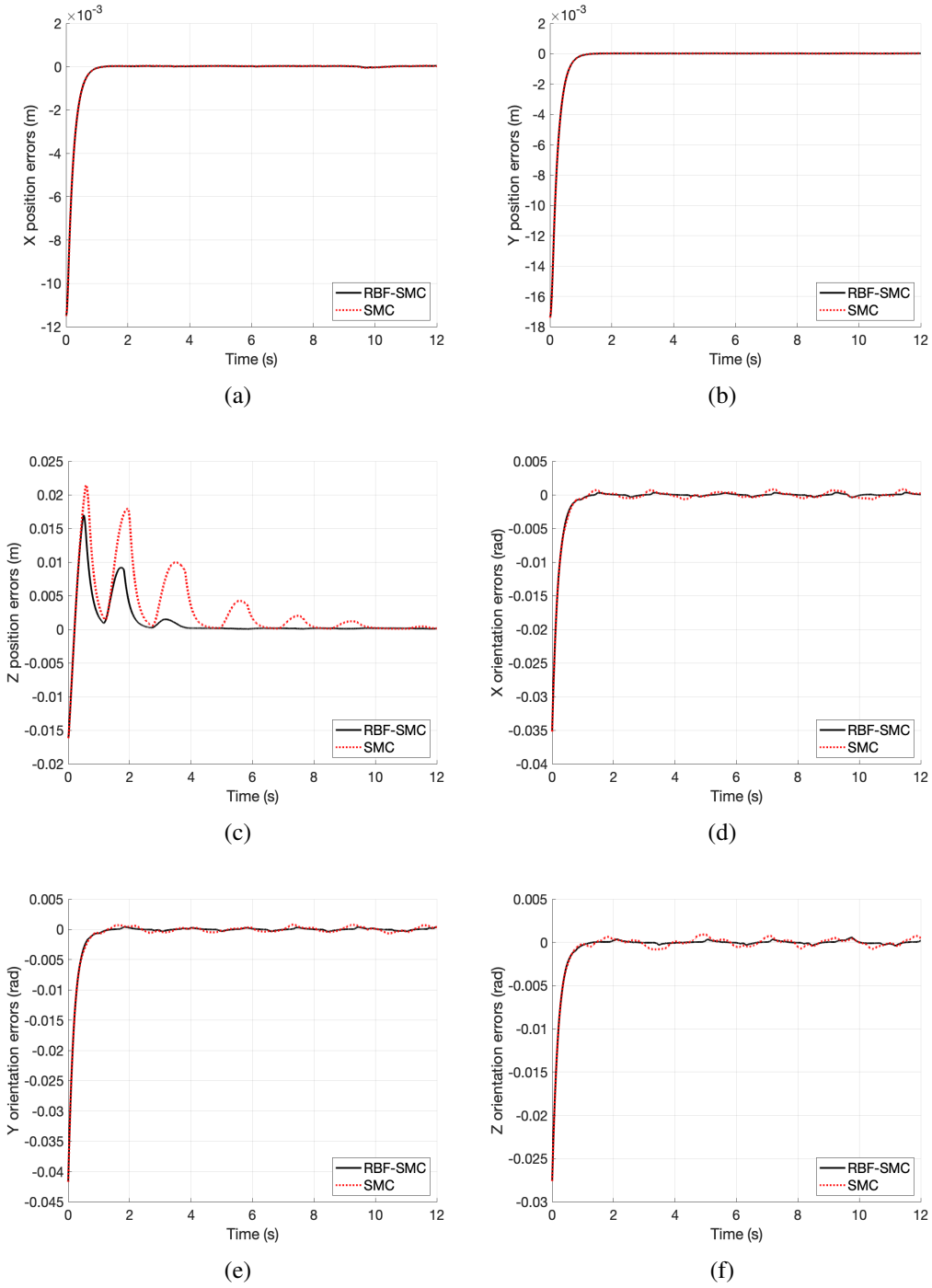


Figure 5.6: Tracking errors of the adaptive sliding mode controller with RBF-neural-network-based tuning method and the sliding mode controller with constant control gains.

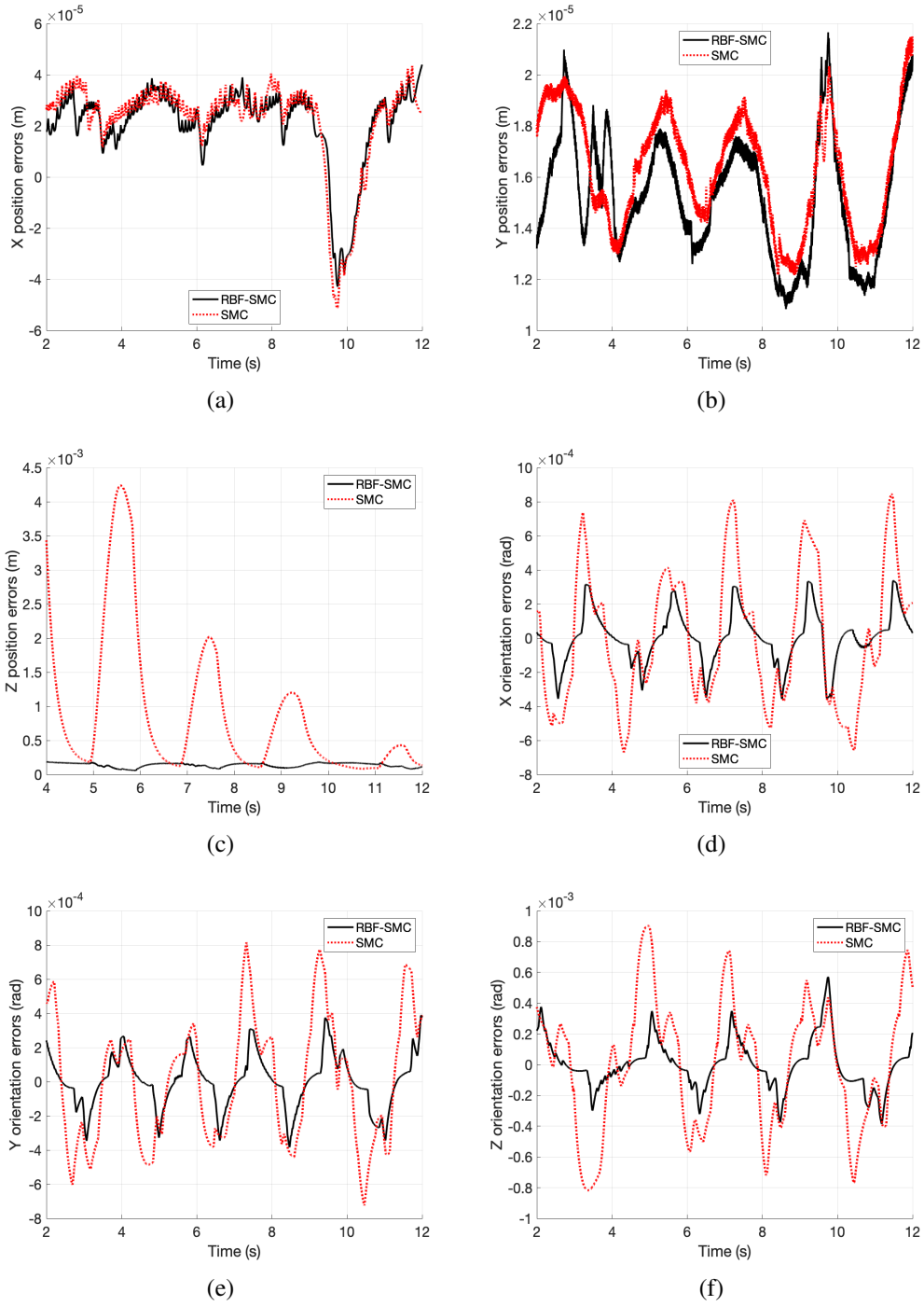


Figure 5.7: Tracking errors of the adaptive sliding mode controller with RBF-neural-network-based tuning method and the sliding mode controller with constant control gains, in the time periods after the tracking convergence.

which is given as

$$X_{d_i}(t) = \sum_{j=1}^{n} \left(\frac{\sin(\omega_0 jt)}{\omega_0 j} s_i^j - \frac{\cos(\omega_0 jt)}{\omega_0 j} c_i^j \right) + X_{0_i} \tag{5.47}$$

where X_{d_i} denotes the desired trajectory in the i-th column, the fundamental frequency ω_0 is set as 0.1 rad/s, and n is the harmonic number and defined as 5. s_i^j, c_i^j, and X_{0_i} are known as trajectory parameters. The initial end-effector pose is chosen as $[0(\mathrm{m}), 0(\mathrm{m}), 0.118(\mathrm{m}), 0(\mathrm{rad}), 0(\mathrm{rad}), 0(\mathrm{rad})]$, and the initial velocity and acceleration are all zero. The initial values of K_p and K_i are set as $\mathrm{diag}([100, 100, 100, 100, 100, 100])$.

The trajectory tracking performance of the two controllers are shown in Figure 5.5, and the tracking errors are illustrated in Figure 5.6. As we can see, both controllers can achieve good tracking performance for the rotational motion in all the three axes and the translational motion along X-axis and Y-axis. It takes longer time for the Z-axis position tracking to converge in both controllers, but the proposed controller can speed up the convergence process. For further comparisons, the tracking errors in the time periods after the tracking convergence are presented in Figure 5.7, and the error indexes including mean absolute error and root mean square error are analyzed in Table 5.1. It can be concluded that the proposed control scheme can greatly reduce the tracking errors for Z position, and the tracking accuracy of rotational motion has also been improved along all the three axes. For the translational motion along X-axis and Y-axis, the two controllers can obtain similar tracking accuracy.

5.6.2 Torque to Position Transformer

In general, only the built-in controller of industrial robots is able to get access to the robot parameters and generate the motion commands. For the 6-RSS parallel robot, the

Table 5.1: Comparison of the error indexes in the simulation

Pose	Indexes	RBF-SMC	SMC
X position (mm)	MAE	0.248	0.25
	RMSE	1.22	1.23
Y position (mm)	MAE	0.373	0.375
	RMSE	1.92	1.93
Z position (mm)	MAE	1.31	3.31
	RMSE	3.26	5.83
X orientation (10^{-3}rad)	MAE	0.7	0.91
	RMSE	3.23	3.26
Y orientation (10^{-3}rad)	MAE	0.8	1
	RMSE	3.79	3.82
Z orientation (10^{-3}rad)	MAE	0.58	0.8
	RMSE	2.58	2.61

built-in controller is designed as PID controller, given as

$$\tau_\theta = k_p e_\theta + k_i \int_0^t e_\theta \, dt + k_d \dot{e}_\theta \tag{5.48}$$

where k_p, k_i and k_d are the control gains, τ_θ is the internal torque signal, $e_\theta = \theta_d - \theta$ is the joint position tracking error, θ_d is the desired joint position, and θ is the actual joint position measured by the potentiometer. In [119], the values of k_p, k_i and k_d have been determined according to the dynamic identification results.

The torque to position transformer can covert the torque signals sent from the designed controller to the position signals the robot can use. Then the torque signals generated by the built-in controller can be the same as the external torque signals. According to [140]

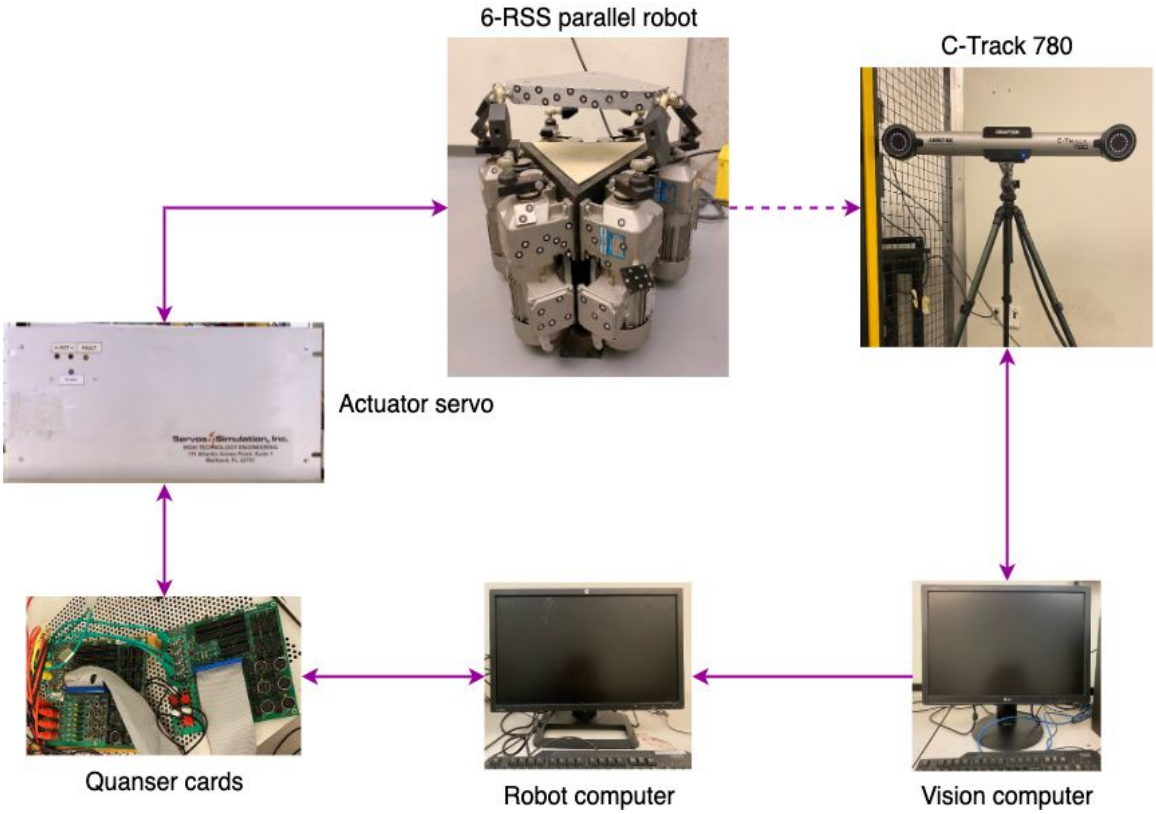


Figure 5.8: Control network of the hardware setup.

and [141], the desired joint position signal θ_d can be obtained as

$$\theta_d = \frac{1}{k_p + k_i p^{-1} + k_d p} \tau_{ex} + \theta \tag{5.49}$$

where τ_{ex} is the external torque signal, and p represents the differential operator.

5.6.3 Experimental Tests

The control network of the experimental setup is presented in Figure 5.8. The parallel robot is modeled in the software VXelements, and twelve reflectors on the upper platform and seven reflectors on the base platform are detected. The base frame and the end-effector frame can be defined by regulating more than three non-collinear reflectors on each platform, and then the end-effector pose can be obtained and tracked. The controller sampling

interval is set as 34.483 ms, which is the same as that of the C-Track 780. Two data acquisition cards Sensoray Model 626 provided by Quanser Inc. are adopted for the joint position communication between the robot and the robot computer. The QUARC software is integrated into MATLAB/Simulink program for real-time control application.

In the experiment, the designed controller is compared with the sliding model controller, and the adaptive sliding mode controllers using single neuron and BP neural network, respectively. The end-effector pose at the initial time is defined as $X = [0(\mathrm{m}), 0(\mathrm{m}), 0.116(\mathrm{m}), 0(\mathrm{rad}), 0(\mathrm{rad}), 0(\mathrm{rad})]$, while the end-effector velocity and acceleration are initialized as 0. The initial learning rates and momentum factor in the RBF neural network are defined as $\mu = 0.2$, $\eta = 0.2$, and $\alpha = 0.05$. They may be updated at each sampling step in order to satisfy the stability conditions in Eqs. (5.32)-(5.35). The initial values of the elements in the width b_n and the center vector c_n in the Gaussian function are defined as 1 and 1.5, respectively. The activation function in BP neural network and single neuron is sigmoid function, and the threshold is set as 0.6. The learning rates in the BP neural network and the single neuron are all set as 0.2. The weights in the three neural networks are all initialized as 1. The initial K_p and K_i are defined as $10^{-3} * \mathrm{diag}([15, 15, 15, 15, 15, 15])$ and $10^{-3} * \mathrm{diag}([8, 8, 8, 8, 8, 8])$, respectively. For the noise covariance matrices in the adaptive Kalman filter, Q_k is initialized as $10^{-5} * \mathrm{diag}([6, 6, 8, 10, 12, 15, 5, 5, 5, 8, 8, 8])$, and S_k is given as $10^{-4} * \mathrm{diag}([2, 2, 2.8, 2.5, 2.5, 3])$. Two different kinds of desired trajectories are applied for the comparisons between the four controllers.

Case 1: The desired trajectory is defined as the sinusoid trajectory with frequency 0.5 rad/s. The tracking errors of the four controllers are illustrated in Figure 5.9. For further evaluation, the error indexes MAE and RMSE are presented in Table 5.2.

Due to the strong robustness of the sliding mode control techniques to system uncertainties, all the controllers can realize good convergence speed and accuracy for trajectory

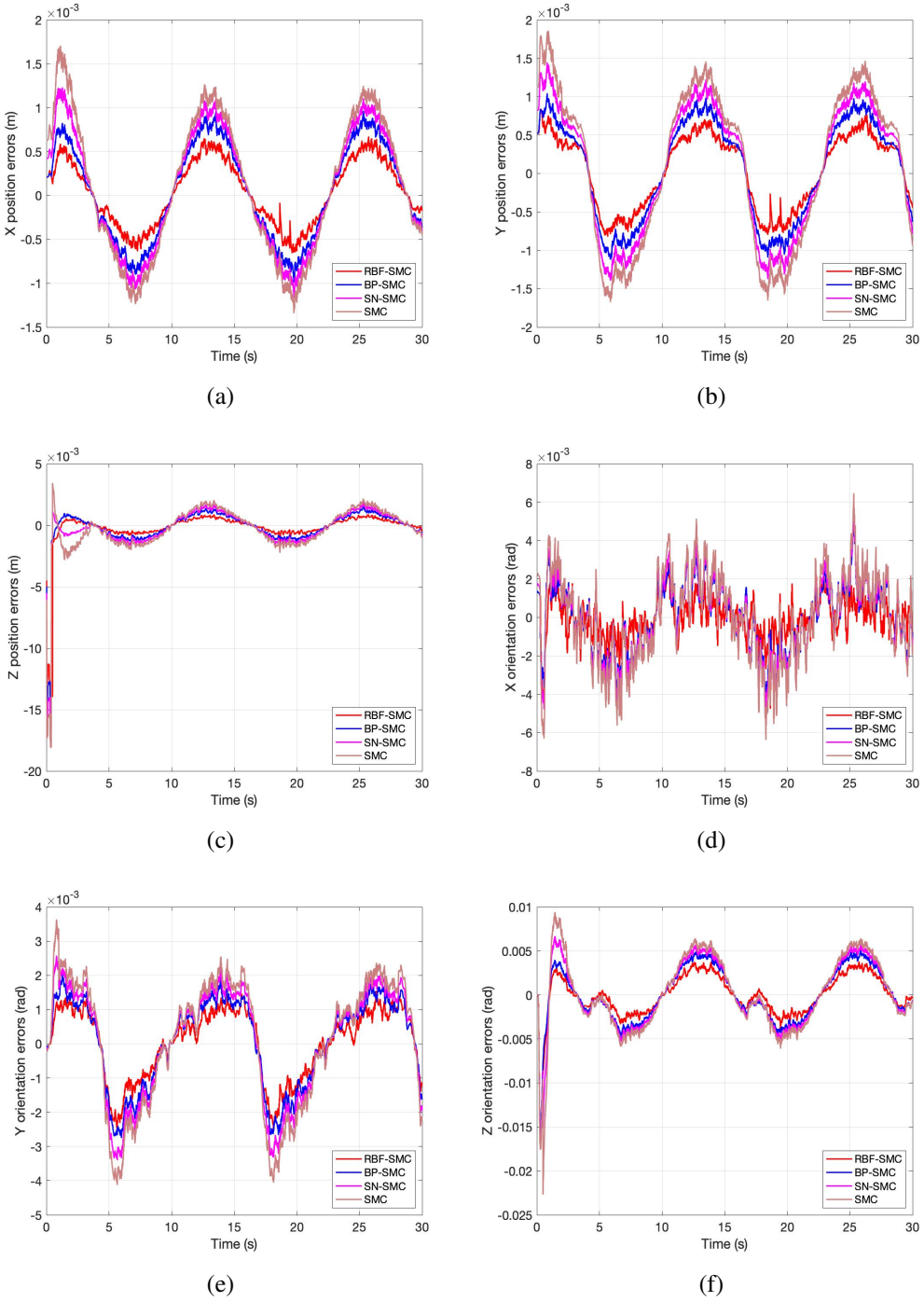


Figure 5.9: Trajectory tracking errors in Case 1. SMC: sliding mode controller; RBF-SMC: adaptive SMC with RBF neural network; BP-SMC: adaptive SMC with BP neural network; SN-SMC: adaptive SMC with single neuron.

Table 5.2: Comparison of the error indexes in Case 1

Pose	Indexes	RBF-SMC	BP-SMC	SN-SMC	SMC
X position (mm)	MAE	0.31	0.48	0.56	0.65
	RMSE	0.38	0.54	0.61	0.78
Y position (mm)	MAE	0.44	0.57	0.71	0.92
	RMSE	0.48	0.64	0.8	0.99
Z position (mm)	MAE	0.62	0.85	1.06	1.35
	RMSE	1.64	1.89	2.34	3.0
X orientation (10^{-3}rad)	MAE	0.82	1.37	1.67	2.1
	RMSE	1.1	1.71	1.93	2.48
Y orientation (10^{-3}rad)	MAE	0.86	1.13	1.34	1.57
	RMSE	1.08	1.32	1.53	1.88
Z orientation (10^{-3}rad)	MAE	1.74	2.42	2.57	2.9
	RMSE	2.36	3.08	3.42	3.87

tracking. By employing neural networks to automatically tune the control gains, the tracking errors of the three adaptive sliding controllers are smaller for all the six dimensions than the sliding mode controller. Moreover, the proposed controller using RBF neural network can obtain higher tracking accuracy compared with the other two adaptive sliding mode controllers. The orientation tracking errors are larger and the responses are with more oscillations compared to the position tracking along all the three axes. The reason is that more sensor noises appear in the orientation measurements than the position measurements. Different from X position and Y position tracking, there exist large transient errors at the initial stage of Z position tracking, which could be caused by the robot modeling errors in VXelements. The X-axis and Y-axis of the base frame and the end-effector frame can be easily defined since the upper platform and the base platform are with symmetrical structures. The initial end-effector positions w.r.t. the base frame along X-axis and Y-axis

are both set as zero. Based on the kinematic calibration results in [119], the modeling value is set as 0.116 m for the initial Z position w.r.t. the base frame, which may cause a larger deviation from the actual value than X position and Y position.

Case 2: To further test the robustness of the developed controller, the finite Fourier series-based trajectory with varying frequency is selected as the desired trajectory, given as Eq. (5.47). The fundamental frequency ω_0 is set as 0.12 rad/s, and the harmonic number n is given as 5. The tracking errors of the four controllers in Case 2 are shown in Figure 5.10. The error indexes MAE and RMSE are analyzed in Table 5.3.

With the sliding mode control techniques, good tracking performance can be achieved for all the controllers in the presence of system uncertainties and varying trajectory frequency. Similar to Case 1, the tracking accuracy of the sliding mode controller is lower than the three adaptive sliding mode controllers. It is clear that with the RBF-neural-network-based auto-tuning approach, the proposed controller can obtain the smallest tracking errors for all the six dimensions. Besides, more oscillations and larger tracking errors still appear in the orientation tracking than the position tracking. The large transient tracking errors for Z position at the initial stage also exist due to the robot modeling errors in VXelements. Different from Case 1, the tracking errors in Case 2 increase when it comes to the high frequency part of the desired trajectory, which could be caused by the fact that the uncertain frictions and noises may increase greatly with the growing velocity. In the case of higher frequency, the designed adaptive sliding mode controller exhibits stronger robustness with smaller transient tracking errors.

For the adaptive sliding mode controller using single neuron, the approximation performance is inevitably sacrificed due to the limited number of neurons. BP neural network is a global approximator since each hidden node has equal influence on the output, but it may encounter the issue of local minima and normally takes longer time for training with a complicated structure. In RBF neural network, the response of a hidden node is

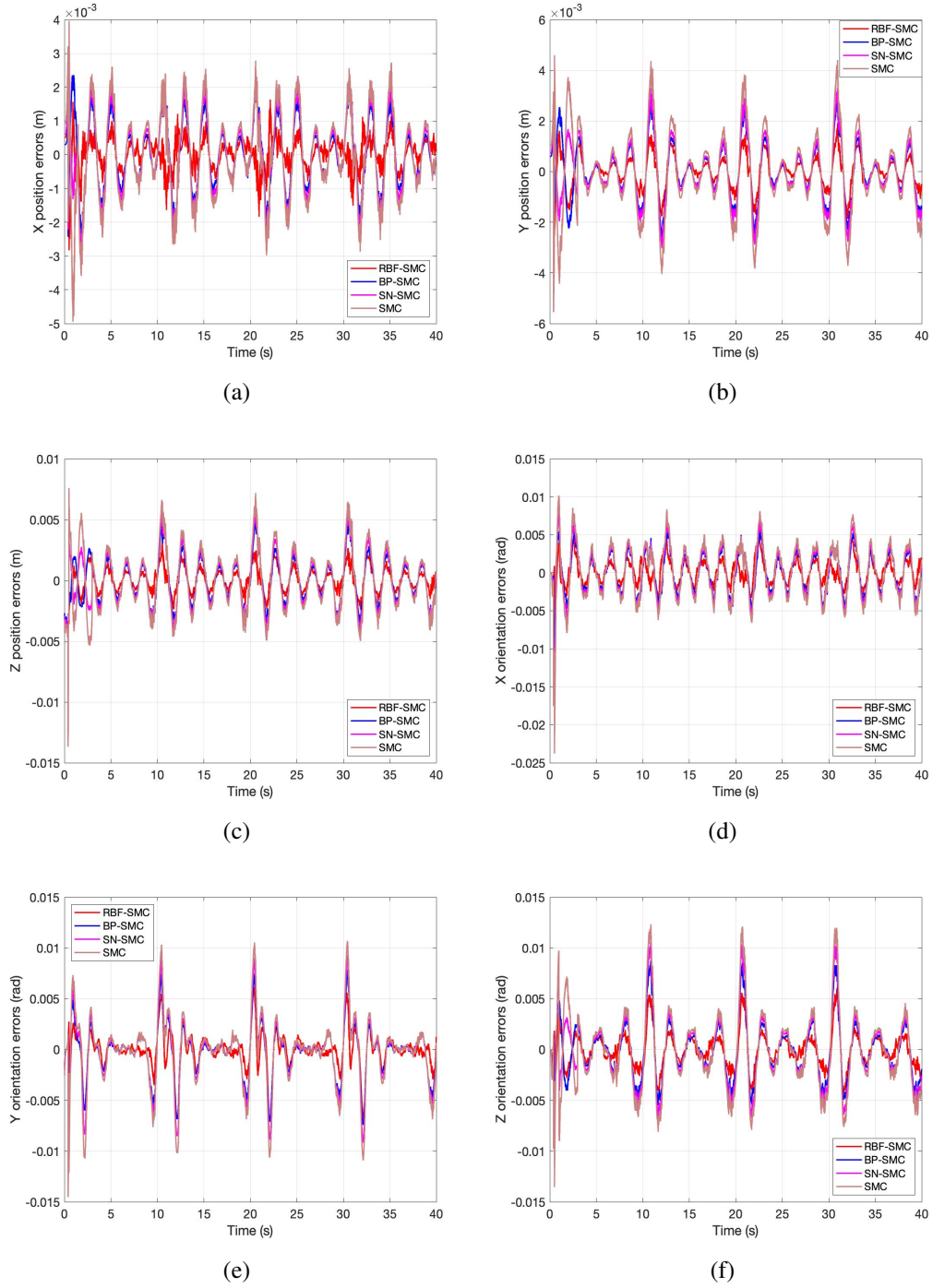


Figure 5.10: Trajectory tracking errors in Case 2. SMC: sliding mode controller; RBF-SMC: adaptive SMC with RBF neural network; BP-SMC: adaptive SMC with BP neural network; SN-SMC: adaptive SMC with single neuron.

Table 5.3: Comparison of the error indexes in Case 2

Pose	Indexes	RBF-SMC	BP-SMC	SN-SMC	SMC
X position (mm)	MAE	0.37	0.76	0.92	1.13
	RMSE	0.49	0.91	1.12	1.37
Y position (mm)	MAE	0.5	0.77	0.96	1.18
	RMSE	0.63	0.98	1.24	1.53
Z position (mm)	MAE	0.83	1.44	1.71	2.0
	RMSE	1.05	1.75	1.98	2.4
X orientation (10^{-3}rad)	MAE	1.32	2.11	2.48	2.9
	RMSE	1.65	2.46	2.92	3.47
Y orientation (10^{-3}rad)	MAE	1.2	1.58	1.87	2.24
	RMSE	1.81	2.39	2.85	3.33
Z orientation (10^{-3}rad)	MAE	1.3	2.05	2.43	2.9
	RMSE	1.67	2.6	3.07	3.74

larger when the Euclidean distance between the input and the center vector of the activation function is smaller. The output of RBF neural network is more related to the hidden nodes where the inputs are closer to the center vector, thus RBF neural network works as a local approximator. Compared with single neuron and BP neural network, RBF neural network demonstrates stronger approximation ability with higher learning accuracy and speed. Therefore, the proposed adaptive sliding mode controller can obtain smaller tracking errors and stronger robustness to system uncertainties in the two experimental tests.

In addition, the root mean square (RMS) values of the control signals in the four controllers are calculated, as given in Figure 5.11, which indicate how energetically expensive of all the controllers. The sliding mode controller can obtain the smallest RMS values due to its simplicity. For the three adaptive sliding mode controllers, the one using single neuron realizes smaller RMS control action values because of the simpler neural network

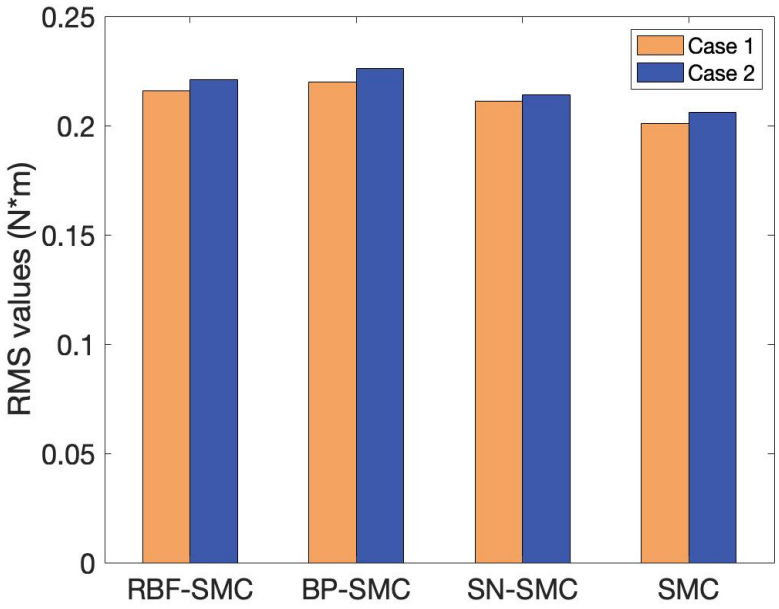


Figure 5.11: RMS values of the control signals in the two cases.

structure. The RMS values of the controller using BP neural network are higher than those of the controller with RBF neural network, since the BP neural network owns more complex structure, which causes larger computational cost of the training process.

5.7 Summary

In this chapter, the position-based visual servoing of the 6-RSS parallel robot is investigated. The photogrammetry sensor C-Track 780 together with the adaptive Kalman filter is able to achieve the accurate end-effector pose estimation. An RBF-neural-network-based adaptive sliding mode controller is developed to cope with system uncertainties and tune the control gains automatically. Detailed stability analysis of the controller utilizing Lyapunov theorem has been provided. The effectiveness of the proposed PBVS approach has been validated by experiments. Simulation and experimental results illustrate that in terms of robustness and tracking accuracy, the designed controller presents better performance

compared with the sliding mode controller and the adaptive sliding mode controllers using BP neural network or single neuron.

Chapter 6

Distributed DRNN-based Adaptive Sliding Mode Control of the Cooperative Robotic System

6.1 Problem Formulation

In model-based sliding mode controllers, the system uncertainties including modeling errors, frictions and external disturbances can degrade the trajectory tracking performance of the robotic systems. To address this issue, some intelligent algorithms have been employed for the approximation of the system uncertainties.

Neural network serves as one of most popular techniques among all the intelligent algorithms [142]. It is equipped with strong nonlinear learning and approximation abilities for the compensation of the system uncertainties in robotic systems. Deep recurrent neural network, which take advantages of both deep neural network and recurrent neural network, has attracted much attention in recent years. By adopting multiple hidden layers and extra

feedback loops in the neural network structure, deep recurrent neural network can demonstrate more powerful dynamic representation and estimation capabilities. It can store and process the dynamic information of the previous steps, and consumes less computation resources than shallow feedforward neural networks with substantial neurons, for complex dynamic system estimation.

As described in Chapter 2, sliding mode control and neural network control are broadly applied in the control of diverse robotic systems. Researchers have also explored integrating sliding mode control with deep recurrent neural networks for the control of nonlinear systems. As presented in [143], a recurrent neural network with two hidden layers is employed to estimate the system uncertainties in a global sliding mode controller for three-phase active power filter system. In [144], a terminal sliding mode control scheme utilizing a deep fuzzy recurrent neural network is developed. It is validated on second-order inverted pendulum system and active power filter system, respectively. Nevertheless, it is worth noting that the nonlinear systems investigated in [143] and [144] are simple systems with low DOFs. In [145], a robust adaptive sliding mode control strategy is proposed for a 3-link industrial robot manipulator. It relies on a deep recurrent fuzzy wavelet neural network to approximate the control terms that encompass the dynamic parameters of the robot. As described in [146], the lumped dynamic uncertainties in a second-order non-singular fast terminal sliding mode controller for a 2-link manipulator, are estimated through a deep fuzzy wavelet neural network incorporated with multiple feedback loops. However, due to the presence of substantial feedback loops within the neural network structure and the absence of experimental validation, it raises concerns about the potential compromise in approximation accuracy and efficiency in real applications.

It can be seen that the DRNN-based adaptive sliding mode control techniques have not been fully investigated for robotic systems, especially when it comes to the robots with high DOFs. In the presence of dynamic uncertainties and external disturbances, this topic is

worth exploring for the 6-DOF serial robot and the 6-RSS parallel robot in the cooperative AFP system.

6.2 Contributions

To enhance the trajectory tracking performance of the two robots in the AFP system, a DRNN-based adaptive sliding mode control approach is designed to cope with dynamic uncertainties and external disturbances in this chapter. Experimental validations are done on the 13-DOF cooperative AFP system using the distributed control structure described in Chapter 3 and the planned leader-follower trajectories in Chapter 4. The detailed contributions are given as follows:

(1) According to the distributed control structure with event-triggered condition in Chapter 3, one joint-space controller for the serial robot and one task-space controller for the parallel robot are designed, respectively, using the DRNN-based adaptive sliding mode control method.

(2) In the proposed controller, a DRNN with three hidden layers and one feedback loop between the output layer and the input layer, is employed to estimate the lumped system uncertainties. The DRNN can obtain better approximation and dynamic capabilities than traditional shallow feedforward neural networks. The adaptation laws of the neural network parameters are derived using Lyapunov theorem to guarantee the stability of the controller.

(3) Compared with the proposed controller in [146], the DRNN-based ASMC is more implementable for high-DOF system by adding one more hidden layer and reducing the number of feedback loops. In addition to estimating the lumped system uncertainties through DRNN, the nominal dynamic parameters are also incorporated into the controller, which makes it more reliable than the control approach outlined in [145]. Moreover, the kinematics and dynamics of the two robots in the AFP system are much more intricate than those of the nonlinear systems in [143] and [144].

(4) Simulation and experiment have been conducted to demonstrate the effectiveness of the proposed distributed control scheme. The advantages of the designed DRNN-based ASMC are presented by making comparisons with the ASMCs using different kinds of neural networks and SMC.

(5) The cooperative control of the 13-DOF AFP system has been investigated in [14] and [104], where only PID controllers are applied. In [14], the proposed synchronized control strategy can lead to complicated communication structure and high workload. In [104], a coordination control approach is developed, with the parallel robot in the leading role and the serial robot in the role of follower. In this chapter, the two robots are controlled in a distributed manner, the parallel robot performs the role of follower and the serial robot is in the role of leader, and the DRNN-based ASMC is designed to enhance the tracking performance of the robots.

The remainder of this chapter is organized as follows. The distributed control structure with the proposed controller is explained first. Then, the models of the serial robot and parallel robot are described. Furthermore, the DRNN-based adaptive sliding mode control approach is introduced, and simulation and comparative experimental results are presented. Finally, a brief summary of this chapter is given.

6.3 Distributed Control Structure Using DRNN-based ASMC

The distributed control structure using the DRNN-based adaptive sliding mode control approach for the cooperative robotic system is demonstrated in Figure 6.1. For the trajectory planning in Chapter 4, the serial robot trajectory is planned in joint space to avoid joint constraints and singularities, while the trajectory of the parallel robot is generated in task space to compensate the motion of the serial robot. Therefore, a joint-space controller

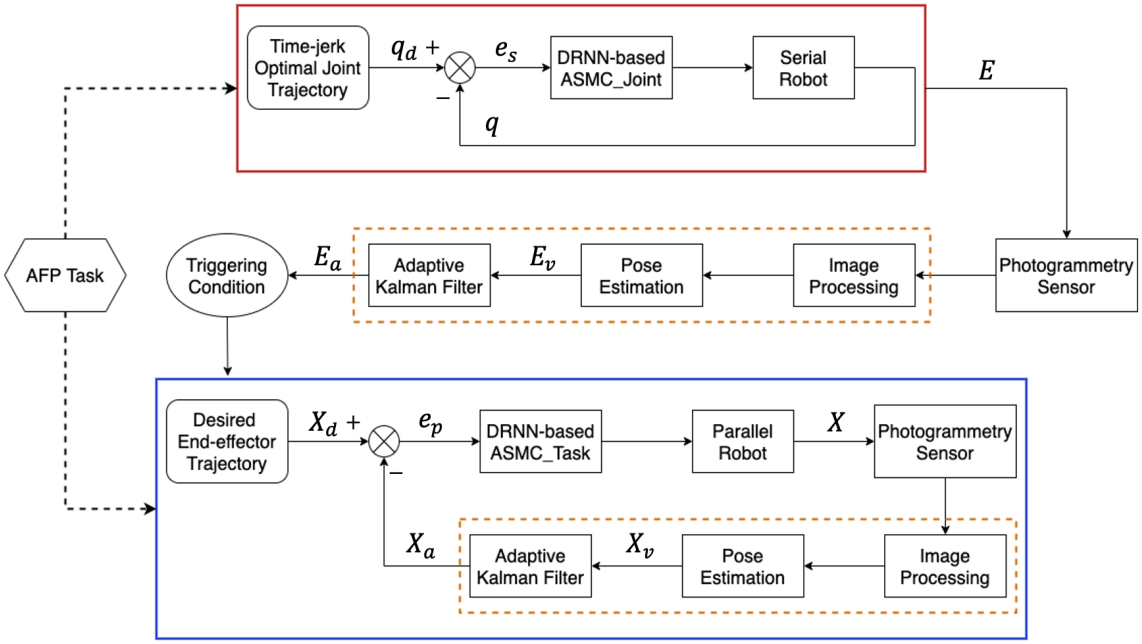


Figure 6.1: Distributed control structure of the cooperative robotic system using DRNN-based ASMC.

for the serial robot and a task-space controller for the parallel robot are developed, respectively. Besides, the position-based visual servoing technique is utilized for the task-space controller design. Moreover, in order to deal with the uncertain noises, the adaptive Kalman filter is utilized to improve the pose estimation accuracy of the two robots.

The definitions of the notations in Figure 6.1 are introduced as follows. For the serial robot, q_d: desired joint position; q: real-time joint position; e_s: joint position error; E: real-time end-effector pose; E_v: measured end-effector pose by photogrammetry sensor; E_a: filtered end-effector pose by AKF. As for the parallel robot, X_d: desired end-effector pose; X: real-time end-effector pose; X_v: measured end-effector pose by photogrammetry sensor; X_a: filtered end-effector pose by AKF; e_p: end-effector pose error. The pre-planned leader-follower trajectories serve as the desired trajectories for the cooperative robotic system. For a continuous fiber layup process on Branches 1 and 2 with 0° fiber path, the parallel robot is stationary when the AFP head can successfully follow the desired fiber path along Branch 1. Since the motion constraints of the serial robot may be encountered

for Branch 2, the cooperative motion between the two robots should be considered to finish the task. The triggering condition for the parallel robot movement is designed as $E_a = E_{at}$, where E_{at} is the filtered end-effector pose of the serial robot by AKF, at the moment while the AFP head reaches the start point of the fiber path along Branch 2. As long as the triggering condition is satisfied, a triggering signal will be sent to the parallel robot to start its movement.

6.4 Robot Dynamic Models

In this section, the dynamic models of the two robots in the 13-DOF cooperative AFP system are given.

Let's define $q \in R^6$, $\dot{q} \in R^6$ and $\ddot{q} \in R^6$ as joint position, velocity and acceleration, respectively, the dynamics of the serial robot can be described as [10]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_f = \tau - \tau_d \tag{6.1}$$

where $M(q) \in R^{6\times6}$ is the inertia matrix, $C(q,\dot{q}) \in R^{6\times6}$ is the Coriolis and centrifugal matrix, $G(q) \in R^6$ is the gravitational term, $\tau_f \in R^6$ denotes the friction term, $\tau_d \in R^6$ denotes the external disturbance, and $\tau \in R^6$ denotes the control input. The dynamic uncertainties can be represented as

$$\begin{aligned} \Delta M &= M(q) - M_0(q) \\ \Delta C &= C(q,\dot{q}) - C_0(q,\dot{q}) \\ \Delta G &= G(q) - G_0(q) \\ \Delta \tau_f &= \tau_f - \tau_{f0}. \end{aligned} \tag{6.2}$$

where $M_0(q)$, $C_0(q,\dot{q})$, $G_0(q)$, and τ_{f0} are the nominal dynamic parameters.

The three properties of the robot dynamic equation are listed below.

Property 1: $M(q)$ is symmetric and uniformly positive definite [10].

Property 2: $\dot{M}(q) - 2C(q, \dot{q})$ is a skew-symmetric matrix. $\forall\ x \in R^6$, $\exists\ x^T(\dot{M}(q) - 2C(q, \dot{q}))x = 0$ [10].

Property 3: τ_d is assumed bounded as $\|\tau_d\| \leq \tau_e$, where τ_e is a positive constant.

Similarly, by denoting $X \in R^6$, $\dot{X} \in R^6$ and $\ddot{X} \in R^6$ as end-effector pose, velocity and acceleration, respectively, the dynamics of the parallel robot can be represented as

$$M'(X)\ddot{X} + C'(X, \dot{X})\dot{X} + G'(X) + \tau_f' = \tau' - \tau_d' \tag{6.3}$$

where $M'(X) \in R^{6\times6}$, $C'(X, \dot{X}) \in R^{6\times6}$, $G'(X) \in R^6$, $\tau_f' \in R^6$, $\tau_d' \in R^6$, and $\tau' \in R^6$.

The objective of this chapter is proposing effective dynamics-based nonlinear controllers to improve the trajectory tracking performance of the two robots. The principle of the designed DRNN-based adaptive sliding mode control approach will be illustrated by introducing the joint-space controller for the serial robot, and the introduction to the task-space controller for the parallel robot will be omitted. Both controllers will be tested by simulation and experiment.

6.5 DRNN-based Adaptive Sliding Mode Control

The structure of the adaptive sliding mode control approach using DRNN for the serial robot is illustrated in Figure 6.2. A DRNN is designed to approximate the lumped system uncertainties within a model-based sliding mode controller.

6.5.1 Sliding Mode Controller

The model-based sliding mode controller is proposed to cope with dynamic uncertainties and external disturbances in the robotic system. The joint position and velocity tracking

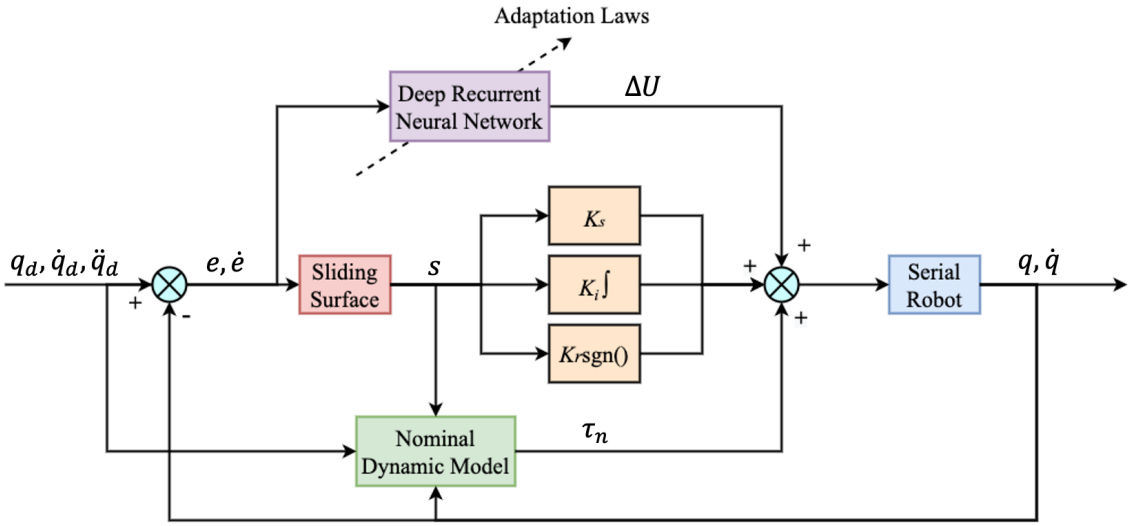


Figure 6.2: DRNN-based adaptive sliding mode control approach.

errors of the serial robot are described as

$$\begin{aligned} e &= q_d - q \\ \dot{e} &= \dot{q}_d - \dot{q} \end{aligned} \tag{6.4}$$

where q_d and $\dot{q}_d$ represent the desired joint position and velocity, respectively, and q and $\dot{q}$ denote the actual joint position and velocity, respectively.

The sliding surface is denoted as

$$s = \dot{e} + \zeta e \tag{6.5}$$

where $\zeta \in R^{6\times 6}$ is a positive definite diagonal matrix.

For further controller design, let's define

$$\begin{aligned} \dot{q}_r &= s + \dot{q} = \dot{q}_d + \zeta e \\ \ddot{q}_r &= \dot{s} + \ddot{q} = \ddot{q}_d + \zeta \dot{e}. \end{aligned} \tag{6.6}$$

Then the sliding mode controller can be developed as

$$\begin{aligned}\tau =& M\left(q\right)\ddot{q}_r + C\left(q,\dot{q}\right)\dot{q}_r + G\left(q\right) + \tau_f + \tau_d + K_s s \\ &+ K_i \int_0^t s\, dt + K_r \mathrm{sgn}(s) \\ =& \tau_n + \Delta U + K_s s + K_i \int_0^t s\, dt + K_r \mathrm{sgn}(s)\end{aligned} \tag{6.7}$$

where $K_s \in R^{6\times 6}$, $K_i \in R^{6\times 6}$ and $K_r \in R^{6\times 6}$ are all positive diagonal matrices, and

$$\tau_n = M_0\left(q\right)\ddot{q}_r + C_0\left(q,\dot{q}\right)\dot{q}_r + G_0\left(q\right) + \tau_{f0}, \tag{6.8}$$

$$\Delta U = \Delta M\ddot{q}_r + \Delta C\dot{q}_r + \Delta G + \Delta\tau_f + \tau_d, \tag{6.9}$$

$$\mathrm{sgn}(s_n) = \begin{cases} 1, & |s_n| > 0 \\ 0, & |s_n| = 0 \\ -1, & |s_n| < 0 \end{cases} \tag{6.10}$$

where $n = 1, 2, ..., 6$.

6.5.2 Deep Recurrent Neural Network Structure

The proposed DRNN in this chapter is depicted in Figure 6.3. Compared with shallow neural networks, which need a significant number of neurons for complex function approximation, the deep neural network incorporating three hidden layers possesses stronger learning ability with less computational consumption. The input layer in the DRNN can receive the feedback signals from the output layer with time delays. The presence of the feedback loop enhances the dynamic capabilities of the neural network structure. By taking advantages of both deep neural network and recurrent neural network, the system can achieve superior approximation performance. The DRNN is adopted to estimate the lumped system uncertainty ΔU in Eq. (6.9). The number of nodes in each hidden layer is set as 6,

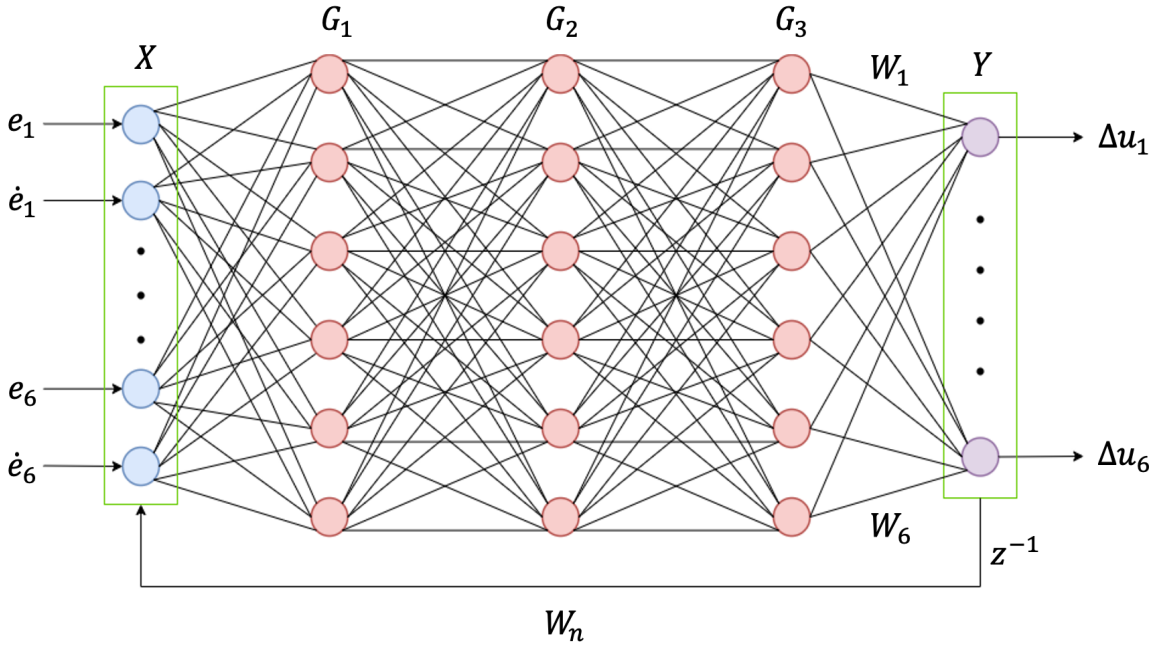


Figure 6.3: Deep recurrent neural network structure.

respectively. All the activation functions in the hidden layers are Gaussian function. The different layers of the DRNN are introduced as follows.

(a) Input layer

The input layer receives two kinds of signals: input signal $X = [e_1, \dot{e}_1, ..., e_6, \dot{e}_6]^T \in R^{12}$ and feedback signal $Y = [\Delta u_1, ..., \Delta u_6]^T \in R^6$ from the output layer associated with the weight $W_n = [W_{n1}, W_{n2}, ..., W_{n6}]^T \in R^{12\times 6}$, where

$$W_n = \begin{bmatrix} W_{n11} & W_{n21} & \cdots & W_{n121} \\ W_{n12} & W_{n22} & \cdots & W_{n122} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n16} & W_{n26} & \cdots & W_{n126} \end{bmatrix}^T . \tag{6.11}$$

The output of the input layer can be formulated as

$$X_1 = X + W_n Y. \tag{6.12}$$

(b) First hidden layer

The output signal of the first hidden layer is given as $G_1 = [G_{11}, ..., G_{16}]^T \in R^6$, and

$$G_{1i} = \exp\left(\frac{-\|X_1 - c_{1i}\|^2}{2b_{1i}^2}\right) \tag{6.13}$$

where $b_{1i} \in R$, $c_{1i} \in R^{12}$, and $i = 1, 2, ..., 6$. The width and the center vector of the nodes in the first hidden layer are described as $b_1 = [b_{11}, b_{12}, ..., b_{16}]^T$ and $c_1 = [c_{11}, c_{12}, ..., c_{16}]^T$, respectively.

(c) Second hidden layer

The output signal of the second hidden layer is defined as $G_2 = [G_{21}, ..., G_{26}]^T \in R^6$, and

$$G_{2j} = \exp\left(\frac{-\|G_1 - c_{2j}\|^2}{2b_{2j}^2}\right) \tag{6.14}$$

where $b_{2j} \in R$, $c_{2j} \in R^6$, and $j = 1, 2, ..., 6$. The width and the center vector of the nodes in the second hidden layer are formulated as $b_2 = [b_{21}, b_{22}, ..., b_{26}]^T$ and $c_2 = [c_{21}, c_{22}, ..., c_{26}]^T$, respectively.

(d) Third hidden layer

The output signal of the third hidden layer is given as $G_3 = [G_{31}, ..., G_{36}]^T \in R^6$, and

$$G_{3m} = \exp\left(\frac{-\|G_2 - c_{3m}\|^2}{2b_{3m}^2}\right) \tag{6.15}$$

where $b_{3m} \in R$, and $c_{3m} \in R^6$, and $m = 1, 2, ..., 6$. The width and the center vector of the nodes in the third hidden layer are denoted as $b_3 = [b_{31}, b_{32}, ..., b_{36}]^T$ and $c_3 = [c_{31}, c_{32}, ..., c_{36}]^T$, respectively.

(e) Output layer

The output of the DRNN is a linear combination of the output of the third hidden layer

with the weight $W = [W_1, W_2, ..., W_6] \in R^{6\times 6}$, where

$$W = \begin{bmatrix} W_{11} & W_{21} & \cdots & W_{61} \\ W_{12} & W_{22} & \cdots & W_{62} \\ \vdots & \vdots & \ddots & \vdots \\ W_{16} & W_{26} & \cdots & W_{66} \end{bmatrix}. \tag{6.16}$$

The final output of the DRNN is

$$Y = W^T G_3. \tag{6.17}$$

6.5.3 DRNN-based Adaptive Sliding Mode Controller

The estimated system uncertainty $\Delta\hat{U}$ through DRNN is expressed as

$$\Delta\hat{U} = \hat{W}^T \hat{G}_3 \tag{6.18}$$

where $\hat{G}_3$ is the estimated output of the third hidden layer using the estimated neural network parameters $\hat{W}$, $\hat{b}_1$, $\hat{c}_1$, $\hat{b}_2$, $\hat{c}_2$, $\hat{b}_3$, $\hat{c}_3$ and $\hat{W}_n$.

Assuming there exist the optimal neural network parameters W^*, b_1^*, c_1^*, b_2^*, c_2^*, b_3^*, c_3^* and W_n^*, leading to the optimal estimation value of ΔU as $\Delta U^* = W^{*T} G_3^*$, The actual lumped system uncertainty is represented as

$$\Delta U = \Delta U^* + \varepsilon \tag{6.19}$$

where ε is the mapping error, $||\varepsilon|| \leq \varepsilon_t$, and ε_t is a small positive constant.

Then the control law in Eq. (6.7) can be rewritten as

$$\tau = \tau_n + \Delta\hat{U} + K_s s + K_i \int_0^t s\, dt + K_r \mathrm{sgn}(s). \tag{6.20}$$

Therefore, the estimation error for ΔU can be computed as

$$\begin{aligned}\Delta U - \Delta\hat{U} =& W^{*T}G_3^* - \hat{W}^T\hat{G}_3 + \varepsilon \\ =& W^{*T}(\hat{G}_3 + \tilde{G}_3) - \hat{W}^T\hat{G}_3 + \varepsilon \\ =& \tilde{W}^T\hat{G}_3 + \tilde{W}^T\tilde{G}_3 + \hat{W}^T\tilde{G}_3 + \varepsilon \\ =& \tilde{W}^T\hat{G}_3 + \hat{W}^T\tilde{G}_3 + \varepsilon_c\end{aligned} \tag{6.21}$$

where $\tilde{W} = W^* - \hat{W}$, $\tilde{G}_3 = G_3^* - \hat{G}_3$, and $\varepsilon_c = \tilde{W}^T\tilde{G}_3 + \varepsilon$.

6.5.4 Stability Analysis

The stability analysis of the proposed adaptive sliding mode controller using DRNN is conducted based on Lyapunov theorem.

Consider the Lyapunov function as

$$\begin{aligned}V =& \frac{1}{2}s^TMs + \frac{1}{2k_1}\mathrm{tr}(\tilde{W}^T\tilde{W}) + \frac{1}{2k_2}\mathrm{tr}(\tilde{b}_1^T\tilde{b}_1) \\ &+ \frac{1}{2k_3}\mathrm{tr}(\tilde{b}_2^T\tilde{b}_2) + \frac{1}{2k_4}\mathrm{tr}(\tilde{b}_3^T\tilde{b}_3) + \frac{1}{2k_5}\mathrm{tr}(\tilde{c}_1^T\tilde{c}_1) \\ &+ \frac{1}{2k_6}\mathrm{tr}(\tilde{c}_2^T\tilde{c}_2) + \frac{1}{2k_7}\mathrm{tr}(\tilde{c}_3^T\tilde{c}_3) + \frac{1}{2k_8}\mathrm{tr}(\tilde{W}_n^T\tilde{W}_n)\end{aligned} \tag{6.22}$$

where k_1, k_2, k_3, k_4, k_5, k_6, k_7 and k_8 are positive constants.

Taking the Taylor expansion of $\tilde{G}_3$ leads to

$$\begin{aligned}\tilde{G}_3 =& dG_{b_1}\tilde{b}_1 + dG_{b_2}\tilde{b}_2 + dG_{b_3}\tilde{b}_3 + dG_{c_1}\tilde{c}_1 + dG_{c_2}\tilde{c}_2 \\ &+ dG_{c_3}\tilde{c}_3 + + dG_{W_n}\tilde{W}_n + O_r\end{aligned} \tag{6.23}$$

where $dG_{b_1} = \left.\frac{\partial\tilde{G}_3}{\partial b_1}\right|_{b_1=\hat{b}_1}$, $dG_{b_2} = \left.\frac{\partial\tilde{G}_3}{\partial b_2}\right|_{b_2=\hat{b}_2}$, $dG_{b_3} = \left.\frac{\partial\tilde{G}_3}{\partial b_3}\right|_{b_3=\hat{b}_3}$, $dG_{c_1} = \left.\frac{\partial\tilde{G}_3}{\partial c_1}\right|_{c_1=\hat{c}_1}$, $dG_{c_2} = \left.\frac{\partial\tilde{G}_3}{\partial c_2}\right|_{c_2=\hat{c}_2}$, $dG_{c_3} = \left.\frac{\partial\tilde{G}_3}{\partial c_3}\right|_{c_3=\hat{c}_3}$, $dG_{W_n} = \left.\frac{\partial\tilde{G}_3}{\partial W_n}\right|_{W_n=\hat{W}_n}$, $\tilde{b}_1 = b_1^* - \hat{b}_1$, $\tilde{b}_2 = b_2^* - \hat{b}_2$, $\tilde{b}_3 = b_3^* - \hat{b}_3$,

$\tilde{c}_1 = c_1^* - \hat{c}_1$, $\tilde{c}_2 = c_2^* - \hat{c}_2$, $\tilde{c}_3 = c_3^* - \hat{c}_3$, $\tilde{W}_n = W_n^* - \hat{W}_n$, and O_r is a high-order term.

Let

$$\begin{aligned} Q =& \frac{1}{2k_1}\mathrm{tr}(\tilde{W}^T\tilde{W}) + \frac{1}{2k_2}\mathrm{tr}(\tilde{b}_1^T\tilde{b}_1) + \frac{1}{2k_3}\mathrm{tr}(\tilde{b}_2^T\tilde{b}_2) \\ &+ \frac{1}{2k_4}\mathrm{tr}(\tilde{b}_3^T\tilde{b}_3) + \frac{1}{2k_5}\mathrm{tr}(\tilde{c}_1^T\tilde{c}_1) + \frac{1}{2k_6}\mathrm{tr}(\tilde{c}_2^T\tilde{c}_2) \\ &+ \frac{1}{2k_7}\mathrm{tr}(\tilde{c}_3^T\tilde{c}_3) + \frac{1}{2k_8}\mathrm{tr}(\tilde{W}_n^T\tilde{W}_n). \end{aligned} \tag{6.24}$$

From Eqs. (6.2), (6.6), (6.8) and (6.9), Eq. (6.1) can be formulated as

$$\begin{aligned} \tau &= M(\ddot{q}_r - \dot{s}) + C(\dot{q}_r - s) + G + \tau_f + \tau_d \\ &= \tau_n + \Delta U - M\dot{s} - Cs. \end{aligned} \tag{6.25}$$

Based on Eqs. (6.20) and (6.25), we can get

$$M\dot{s} + Cs = \Delta U - \Delta\hat{U} - K_s s - K_i \int_0^t s\, dt - K_r \mathrm{sgn}(s). \tag{6.26}$$

Taking the derivative of V yields

$$\dot{V} = s^T M\dot{s} + \frac{1}{2}s^T\dot{M}s + \dot{Q} = s^T(M\dot{s} + Cs) + \dot{Q}. \tag{6.27}$$

According to Eqs. (6.21), (6.23), (6.24) and (6.26), Eq. (6.27) can be described as

$$\begin{aligned} \dot{V} =& -s^T\left(K_s s + K_i\int_0^t s\, dt + K_r\mathrm{sgn}(s)\right) + s^T\hat{W}^T(dG_{b_1}\tilde{b}_1 + dG_{b_2}\tilde{b}_2 + dG_{b_3}\tilde{b}_3 \\ &+ dG_{c_1}\tilde{c}_1 + dG_{c_2}\tilde{c}_2 + dG_{c_3}\tilde{c}_3 + dG_{W_n}\tilde{W}_n + O_r) + s^T(\tilde{W}^T\hat{G}_3 + \varepsilon_c) \\ &+ \frac{1}{k_1}\mathrm{tr}(\tilde{W}^T\dot{\tilde{W}}) + \frac{1}{k_2}\mathrm{tr}(\dot{\tilde{b}}_1^T\tilde{b}_1) + \frac{1}{k_3}\mathrm{tr}(\dot{\tilde{b}}_2^T\tilde{b}_2) + \frac{1}{k_4}\mathrm{tr}(\dot{\tilde{b}}_3^T\tilde{b}_3) \\ &+ \frac{1}{k_5}\mathrm{tr}(\dot{\tilde{c}}_1^T\tilde{c}_1) + \frac{1}{k_6}\mathrm{tr}(\dot{\tilde{c}}_2^T\tilde{c}_2) + \frac{1}{k_7}\mathrm{tr}(\dot{\tilde{c}}_3^T\tilde{c}_3) + \frac{1}{k_8}\mathrm{tr}(\dot{\tilde{W}}_n^T\tilde{W}_n). \end{aligned} \tag{6.28}$$

Let

$$\begin{cases} s^T\tilde{W}^T\hat{G}_3 + \dfrac{1}{k_1}\mathrm{tr}(\tilde{W}^T\dot{\tilde{W}}) = 0 \\ s^T\hat{W}^T dG_{b1}\tilde{b}_1 + \dfrac{1}{k_2}\mathrm{tr}(\dot{\tilde{b}}_1^T\tilde{b}_1) = 0 \\ s^T\hat{W}^T dG_{b2}\tilde{b}_2 + \dfrac{1}{k_3}\mathrm{tr}(\dot{\tilde{b}}_2^T\tilde{b}_2) = 0 \\ s^T\hat{W}^T dG_{b3}\tilde{b}_3 + \dfrac{1}{k_4}\mathrm{tr}(\dot{\tilde{b}}_3^T\tilde{b}_3) = 0 \\ s^T\hat{W}^T dG_{c1}\tilde{c}_1 + \dfrac{1}{k_5}\mathrm{tr}(\dot{\tilde{c}}_1^T\tilde{c}_1) = 0 \\ s^T\hat{W}^T dG_{c2}\tilde{c}_2 + \dfrac{1}{k_6}\mathrm{tr}(\dot{\tilde{c}}_2^T\tilde{c}_2) = 0 \\ s^T\hat{W}^T dG_{c3}\tilde{c}_3 + \dfrac{1}{k_7}\mathrm{tr}(\dot{\tilde{c}}_3^T\tilde{c}_3) = 0 \\ s^T\hat{W}^T dG_{W_n}\tilde{W}_n + \dfrac{1}{k_8}\mathrm{tr}(\dot{\tilde{W}}_n^T\tilde{W}_n) = 0. \end{cases} \tag{6.29}$$

Thus the adaptation laws of the DRNN parameters can be obtained as

$$\begin{cases} \dot{\tilde{W}} = -k_1\hat{G}_3 s^T \\ \dot{\tilde{b}}_1^T = -k_2 s^T\hat{W}^T dG_{b_1} \\ \dot{\tilde{b}}_2^T = -k_3 s^T\hat{W}^T dG_{b_2} \\ \dot{\tilde{b}}_3^T = -k_4 s^T\hat{W}^T dG_{b_3} \\ \dot{\tilde{c}}_1^T = -k_5 s^T\hat{W}^T dG_{c_1} \\ \dot{\tilde{c}}_2^T = -k_6 s^T\hat{W}^T dG_{c_2} \\ \dot{\tilde{c}}_3^T = -k_7 s^T\hat{W}^T dG_{c_3} \\ \dot{\tilde{W}}_n^T = -k_8 s^T\hat{W}^T dG_{W_n}. \end{cases} \tag{6.30}$$

Then Eq. (6.28) can be rewritten as

$$\dot{V} = -s^T\left(K_s s + K_i\int_0^t s\,dt + K_r\mathrm{sgn}(s) - \varepsilon_c - O_{wr}\right). \tag{6.31}$$

where $O_{wr} = \hat{W}^T O_r$. Assuming ε_c and O_{wr} are bounded as $||\varepsilon_c|| \leq \varepsilon_e$ and $||O_{wr}|| \leq O_e$,

where ε_e and O_e are small positive constants. If $K_r = \mathrm{diag}([K_{r1}, K_{r2}, ..., K_{r6}])$ in Eq. (6.7) satisfies

$$K_{rk} \geq \gamma + |u_{sk}| + |u_{ik}| + \varepsilon_e + O_e \tag{6.32}$$

where $k = 1, 2, .., 6$, γ is a small positive constant, $u_s = K_s s$, and $u_i = K_i \int_0^t s \, dt$, it can lead to

$$\dot{V} \leq -\sum_{k=1}^{6} \gamma \, |s_k| \leq -\gamma (s^T s)^{\frac{1}{2}} \leq 0. \tag{6.33}$$

If the DRNN parameters can converge to the optimal values through the adaption laws indicated in Eq. (6.30), the sliding surface s will keep decreasing. According to *Lemma 1* in Chapter 5, it can be concluded that the sliding surface $s = 0$ will be reached within a finite time, and the stability of the system can be guaranteed.

6.6 Simulation and Experimental Results

6.6.1 Simulation Tests

To verify the effectiveness of the designed DRNN-based adaptive sliding mode control approach, it is tested on the two robots in the simulation, respectively. The desired joint trajectory for the serial robot is chosen as sinusoid trajectory, and the frequency is set as 0.5 rad/s. The desired end-effector trajectory for the parallel robot is selected as finite Fourier series-based trajectory with changing frequency, as indicated in Eq. (5.47).

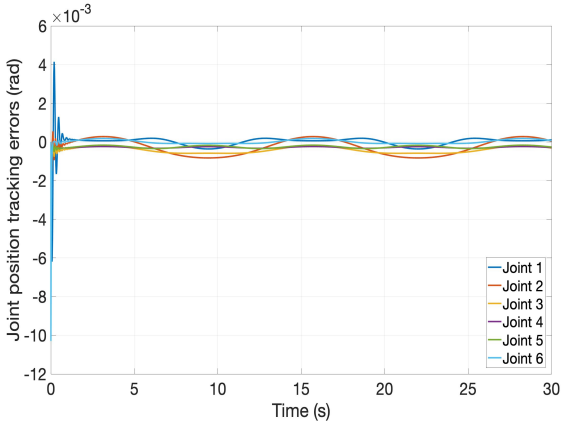
The initial joint position of the serial robot is defined as $[0(\mathrm{rad}), 0(\mathrm{rad}), 0(\mathrm{rad}), 0(\mathrm{rad}), 0(\mathrm{rad}), 0(\mathrm{rad})]^{\mathrm{T}}$, and the initial end-effector pose for the parallel robot is given as $[0(\mathrm{m}), 0(\mathrm{m}), 0.116(\mathrm{m}), 0(\mathrm{rad}), 0(\mathrm{rad}), 0(\mathrm{rad})]^{\mathrm{T}}$. The nominal dynamic parameters are expressed as $M_0 = 0.8M$, $C_0 = 0.8C$, $G_0 = 0.8G$ and $\tau_{f0} = 0.8\tau_f$, respectively. The external disturbance is defined w.r.t. time t as $\tau_d = [2\sin(t), 2\sin(t), 2\sin(t), 2\cos(t), 2\cos(t), 2\cos(t)]^T$. The initial values of the weights

in the DRNN are set as 1, and the elements of the width b and the center vector c in the activation functions are all initialized as 1.2 and 1.5, respectively. In the joint-space controller for the 6-DOF serial robot, K_s and K_i are set as $\mathrm{diag}([180, 180, 180, 180, 180, 180])$ and $\mathrm{diag}([120, 120, 120, 120, 120, 120])$, respectively. The initial value of K_r is chosen as $\mathrm{diag}([50, 50, 50, 50, 50, 50])$. Regarding the task-space controller for the 6-RSS parallel robot, K_s and K_i are given as $\mathrm{diag}([160, 160, 160, 160, 160, 160])$ and $\mathrm{diag}([100, 100, 100, 100, 100, 100])$, respectively. The initial value of K_r is defined as $\mathrm{diag}([40, 40, 40, 40, 40, 40])$.

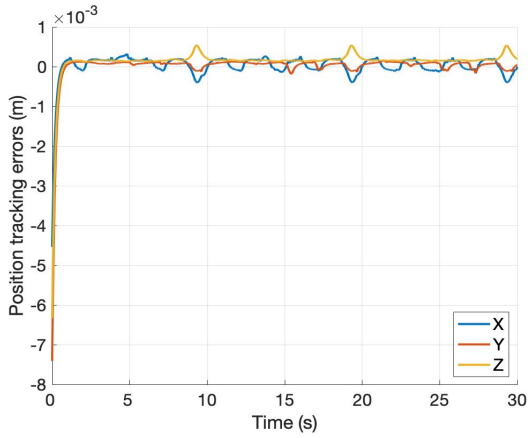
The trajectory tracking performance of the serial robot and parallel robot are demonstrated in Figure 6.4. With the DRNN for system uncertainty estimation and the robust sliding mode control techniques, high tracking accuracy and convergence speed can be realized in both joint-space controller and task-space controller. Besides, as presented in Figure 6.5, the approximated lumped system uncertainties can converge to the optimal values fast with small estimation errors through the designed DRNN in the two simulation tests. It can be concluded that the DRNN-based adaptive sliding mode control approach is able to achieve good trajectory tracking performance for the serial robot and parallel robot, in the presence of dynamic uncertainties and external disturbances.

6.6.2 Experimental Tests

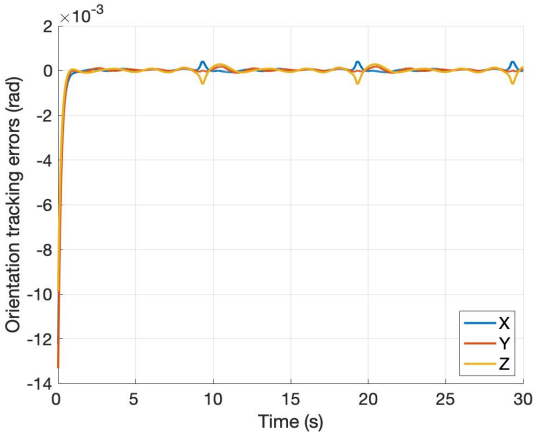
The distributed control strategy using DRNN-based ASMC is validated on the cooperative robotic system in the experiment. The kinematic relationships that need to be satisfied during the cooperative motion include two main points: (1) The compaction roller direction should be kept perpendicular to the surface of the mandrel. (2) A distance defined as $dist = 0.015$ m between the mandrel surface and the roller should be maintained along the roller direction. The desired trajectories of the two robots are the generated leader-follower



(a)

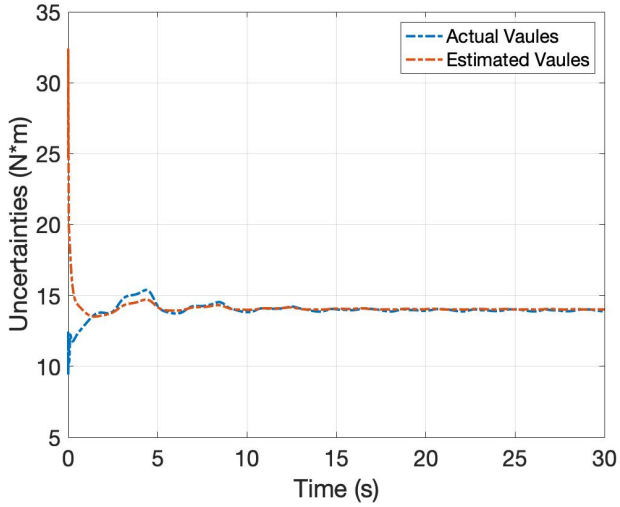


(b)

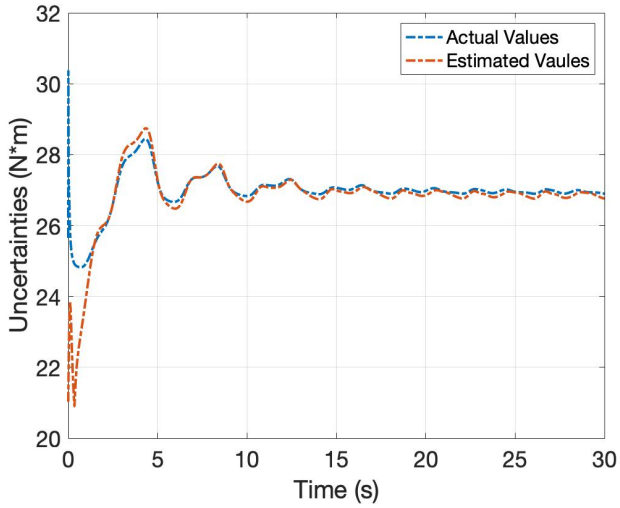


(c)

Figure 6.4: Trajectory tracking errors of the two robots in the simulation. (a) Serial robot joint position. (b) (c) Parallel robot end-effector pose.



(a)

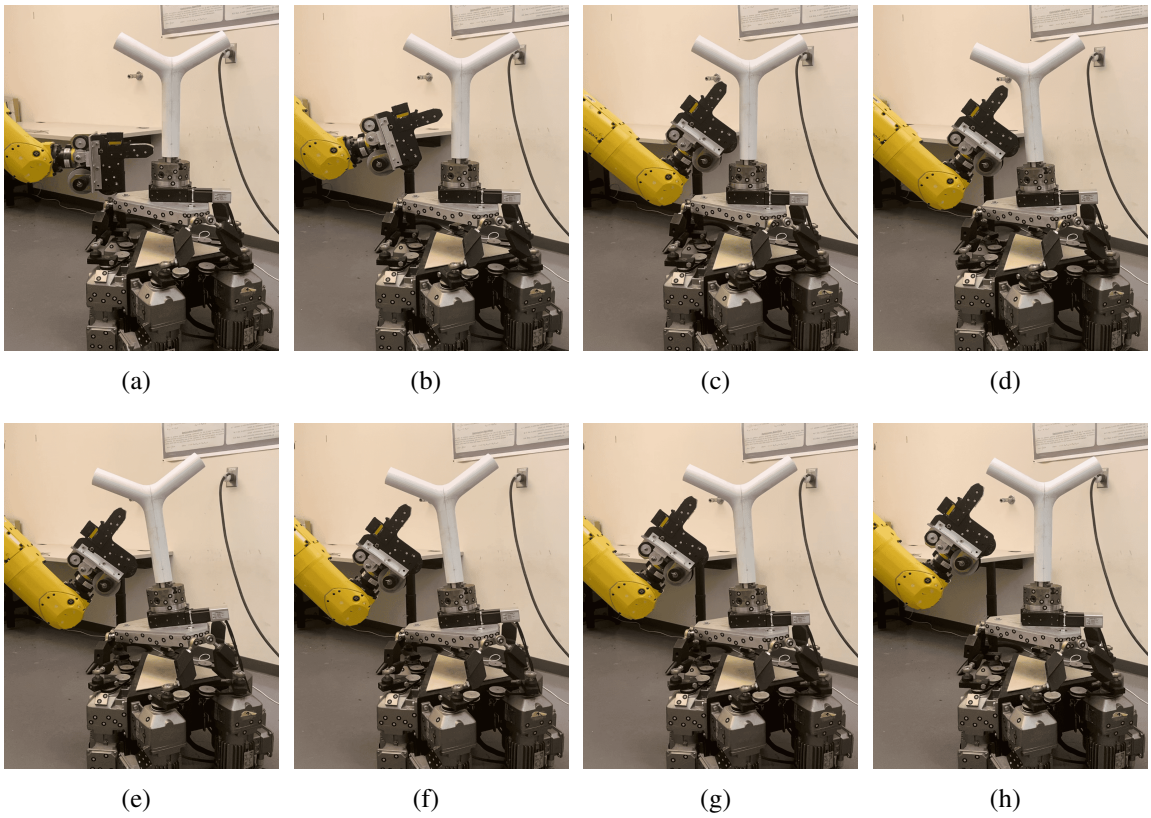


(b)

Figure 6.5: Estimation results of the lumped system uncertainties. (a) Serial robot. (b) Parallel robot.

trajectories in Chapter 4, for a continuous 0° fiber layup on Branches 1 and 2 of the mandrel. Furthermore, comparative experiment is conducted between the following control approaches for the two robots, respectively. In the hidden layers of all the neural network structures, the number of nodes is set as 6.

(a) DRNN-based ASMC developed in this chapter.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6.6: Cooperative motion of the two robots using the distributed DRNN-based adaptive sliding mode control strategy.

(b) ASMC using shallow recurrent neural network (RNN).

(c) ASMC using shallow RBF neural network.

(d) SMC without using neural networks for the estimation of the lumped system uncertainties.

The nominal dynamic parameters of the two robots and the external disturbance in the experiment are defined same as the ones in the simulation. For the parameters in the three neural network structures, the initial values of the weights are set as 1, and the elements in the width b and the center vector c are initialized as 1.5 and 1.8, respectively. Regarding the joint-space controller for the serial robot, K_s and K_i are set as $10^{-2} * \mathrm{diag}([30, 30, 30, 20, 20, 20])$ and $10^{-2} * \mathrm{diag}([12, 12, 12, 6, 6, 6])$, respectively. The

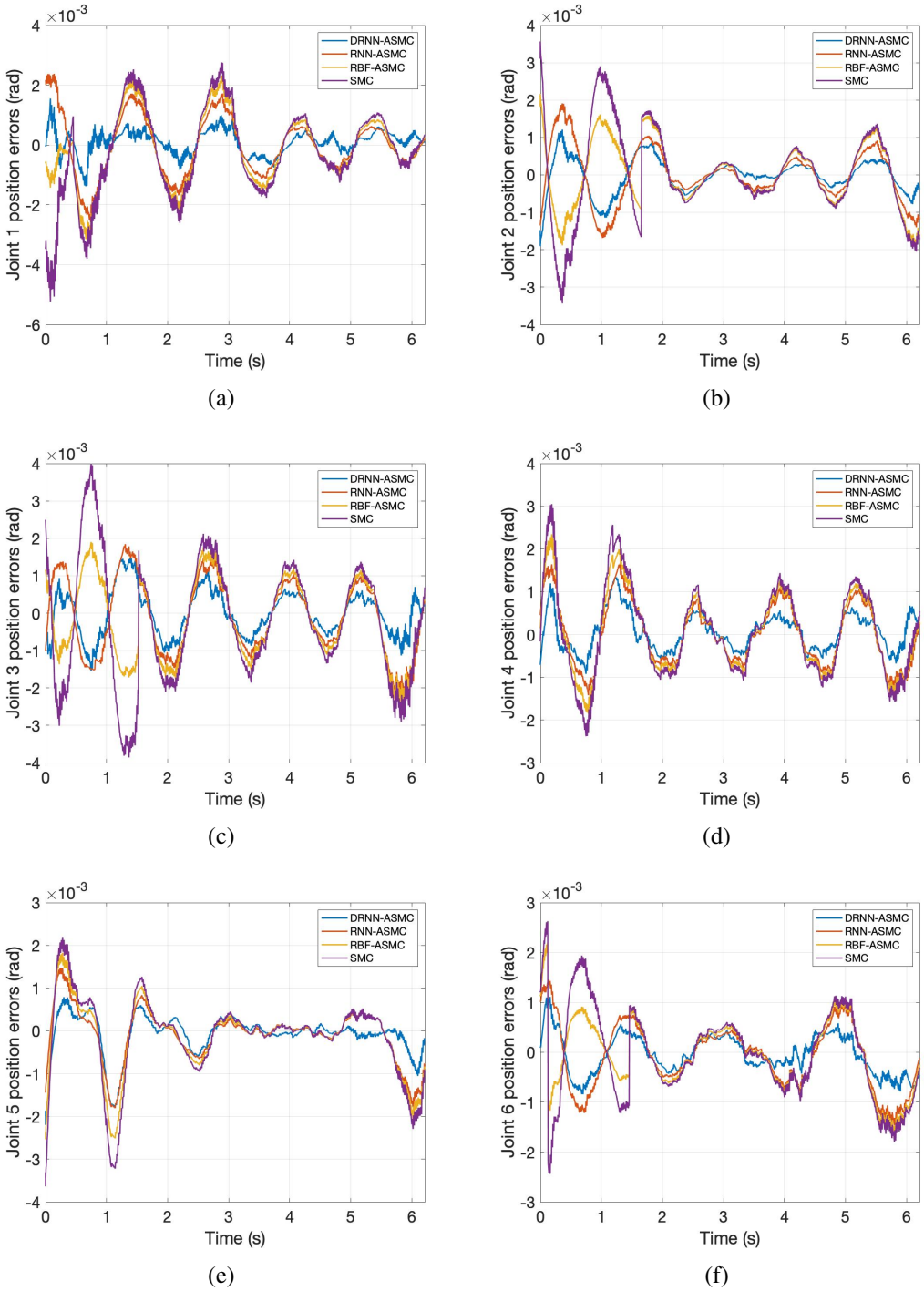


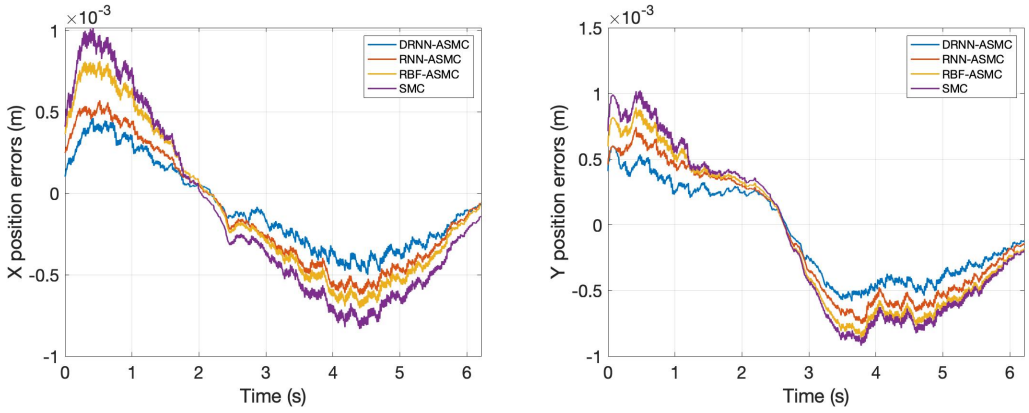
Figure 6.7: Trajectory tracking errors of the serial robot using the three ASMCs with different kinds of neural networks and SMC.

Table 6.1: Error Index Comparison for the Serial Robot

Position (10^{-3}rad)	Indexes	DRNN-ASMC	RNN-ASMC	RBF-ASMC	SMC
Joint 1	MAE	0.4	0.76	0.92	1.07
	RMSE	0.5	0.92	1.12	1.32
Joint 2	MAE	0.42	0.58	0.82	0.91
	RMSE	0.66	0.74	1.04	1.18
Joint 3	MAE	0.68	1.04	1.21	1.5
	RMSE	0.91	1.27	1.46	1.82
Joint 4	MAE	0.44	0.62	0.73	0.85
	RMSE	0.54	0.74	0.88	1.03
Joint 5	MAE	0.38	0.48	0.56	0.65
	RMSE	0.61	0.71	0.85	1.01
Joint 6	MAE	0.42	0.61	0.7	0.78
	RMSE	0.56	0.79	0.91	1.02

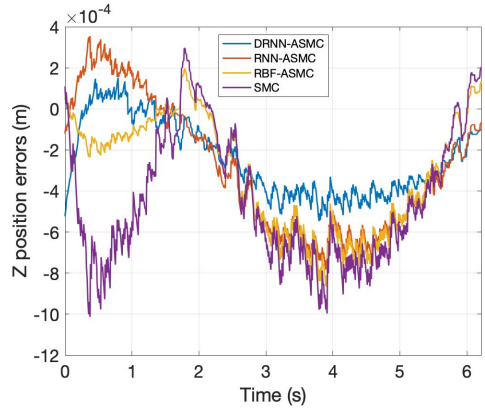
initial value of K_r is defined as $10^{-3} * \mathrm{diag}([50, 50, 50, 40, 40, 40])$. In the task-space controller for the parallel robot, K_s and K_i are defined as $10^{-3} * \mathrm{diag}([20, 20, 20, 20, 20, 20])$ and $10^{-3} * \mathrm{diag}([15, 15, 15, 15, 15, 15])$, respectively. The initial value of K_r is given as $10^{-3} * \mathrm{diag}([10, 10, 10, 10, 10, 10])$.

The cooperative manipulation results using the distributed DRNN-based adaptive sliding mode control strategy are given in Figure 6.6. The parallel robot is stationary as the AFP head follows the desired fiber path along Branch 1. As shown in Figure 6.6 (c), the AFP head reaches the start point of the fiber path along Branch 2. At this moment, the C-Track 780 receives the real-time end-effector pose of the serial robot, which satisfies the triggering condition described in Section 6.3, then the parallel robot starts to move. The two robots move smoothly and continuously without any disruptions, and the AFP kinematic

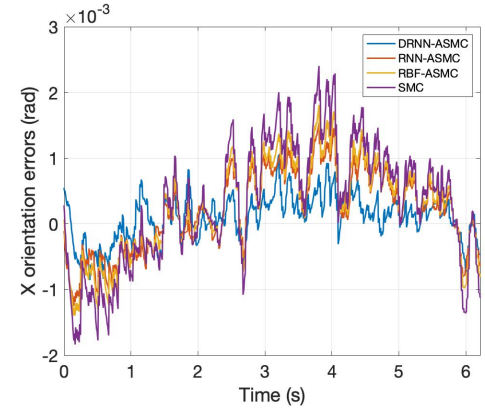


(a)

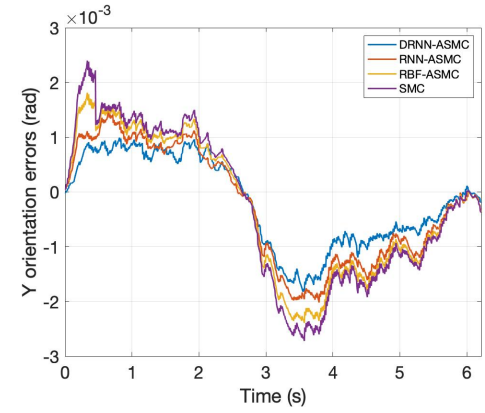
(b)



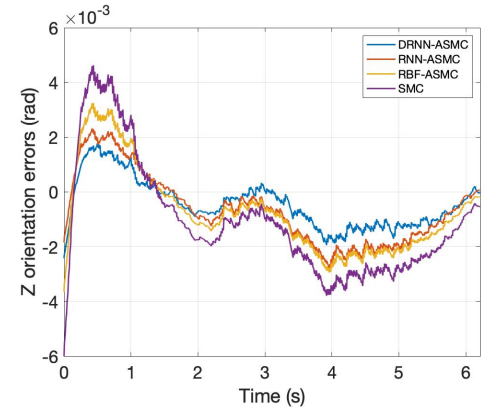
(c)



(d)



(e)



(f)

Figure 6.8: Trajectory tracking errors of the parallel robot using using the three ASMCs with different kinds of neural networks and SMC.

Table 6.2: Error Index Comparison for the Parallel Robot

Pose	Indexes	DRNN-ASMC	RNN-ASMC	RBF-ASMC	SMC
X position (mm)	MAE	0.25	0.33	0.39	0.43
	RMSE	0.29	0.37	0.44	0.48
Y position (mm)	MAE	0.31	0.41	0.46	0.5
	RMSE	0.33	0.45	0.51	0.55
Z position (mm)	MAE	0.32	0.41	0.52	0.63
	RMSE	0.84	0.86	0.92	0.93
X orientation (10^{-3}rad)	MAE	0.31	0.54	0.62	0.81
	RMSE	0.38	0.64	0.74	0.98
Y orientation (10^{-3}rad)	MAE	0.64	0.83	0.96	1.06
	RMSE	0.75	0.89	1.1	1.22
Z orientation (10^{-3}rad)	MAE	1.03	1.44	1.52	1.94
	RMSE	1.39	1.77	1.86	2.4

relationships are maintained simultaneously. It is clearly seen that the distributed control structure with the event-triggered communication network is able to realize the efficient control of the cooperative AFP system.

Regarding the comparative experiment using different control approaches, the trajectory tracking performance of the serial robot and the parallel robot are demonstrated in Figure 6.7 and Figure 6.8, respectively. As given in Table 6.1 and Table 6.2, the error indexes MAE and RMSE are computed, in order to further evaluate and compare the test results. With the strong robustness to dynamic uncertainties and external disturbances using sliding mode control techniques, the four controllers can all achieve high accuracy and convergence speed for trajectory tracking. In comparison to the model-based SMC, smaller

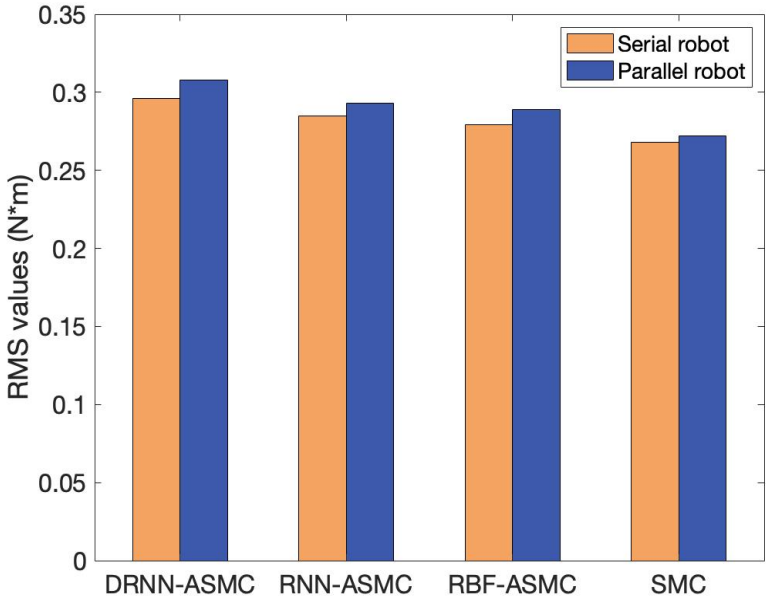


Figure 6.9: RMS values of the control signals for the two robots.

tracking errors can be obtained for the three ASMCs by adopting neural networks to estimate the lumped system uncertainties. The designed DRNN-based ASMC can acquire the highest tracking accuracy among the three ASMCs. The DRNN possesses stronger dynamic approximation ability by combining the advantages of deep neural network and recurrent neural network. We can conclude that the proposed DRNN-based ASMC can realize good trajectory tracking performance and demonstrates strong robustness in dealing with the system uncertainties for the two robots.

As in Chapter 5, the RMS values of the control signals in the four controllers are also computed in this chapter, as presented in Figure 6.9. The model-based sliding mode controller without neural network compensation owns the lowest RMS value. Although superior tracking performance can be obtained in the DRNN-based ASMC, the increase of the number of hidden layers and the extra feedback loop in the DRNN definitely lead to the highest computation consumption.

6.7 Summary

To fulfill the efficient control of the 13-DOF cooperative AFP system, a novel distributed control strategy using DRNN-based ASMC is developed in this chapter. The cooperative control of the serial robot and the parallel robot in the system is conducted in a distributed formation with an event-triggered communication network. An adaptive sliding mode control approach is proposed for the robotic system with the existence of dynamic uncertainties and external disturbances. A DRNN with three hidden layers and a feedback loop from the output layer to the input layer, is designed to approximate the lumped system uncertainties in the controller. The stability analysis of the controller is given according to Lyapunov theorem. The feasibility of the distributed control strategy has been validated by simulation and experiment. The superiority of the DRNN-based ASMC regarding trajectory tracking performance, has been illustrated by the comparisons with ASMCs using different neural networks and SMC.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Aiming to improve the efficiency and accuracy in producing composite components, automated fiber placement system has been extensively applied to manufacture fiber composites in industry. Unfortunately, due to the insufficient system DOFs, the majority of the AFP systems are not able to manufacture composite components with complicated structures. A 13-DOF cooperative robotic system has been developed to deal with this issue, for completing the fiber layup process on a Y-shape mandrel. An AFP head is attached to the end-effector of a 6-DOF serial robot, while the mandrel is held by a 1-DOF rotary stage attached to the end-effector of a 6-RSS parallel robot. A photogrammetry sensor C-Track 780 is adopted for 3D pose estimation in real time.

Efficient control of industrial robots is key to the success of automated manufacturing work, while subject to various motion constraints caused by the complicated kinematic structure and nonlinear dynamics of the robots. The robot control problem is commonly treated by a two-level solution. The first level is trajectory planning, and the second level is trajectory tracking. Trajectory planning computes a time history of the desired positions and velocities. The objective of trajectory tracking is to make the actual values of the

robot's positions and velocities reach the desired values.

The AFP task is more sophisticated than the traditional cooperative manipulation tasks, which focus on either coordinated grabbing of common objects or formation control of multiple robots. The aim of this Ph.D. project is proposing cooperative trajectory planning and control algorithms for the robotic system to place fiber on the Y-shape mandrel with the desired path angle 0°. The research achievements can be summarized as follows.

(1) Distributed control of the cooperative robotic system

Practically, to avoid the excessive communication demands among the two robots, which can degrade system performance and disrupt the overall stability of the network, a distributed control structure with event-triggered condition is developed. Two individual controllers are applied for the control of the serial robot and the parallel robot, respectively. According to the detected end-effector pose of the serial robot by the photogrammetry sensor C-Track 780, the information exchange between the two robots starts solely upon meeting the predefined triggering condition. An adaptive Kalman filter is utilized to estimate the uncertain noises in pose estimation.

(2) Leader-follower trajectory planning for the cooperative robotic system

This work investigates a leader-follower trajectory planning strategy for the two robots. The serial robot with the AFP head is employed as the leader, while the 6-RSS parallel robot holding the Y-shape mandrel serves as the follower. Due to the fact that the dynamic and kinematic constraints of the serial robot could disrupt the fiber layup process, a time-jerk optimal trajectory planning method is formulated for the serial robot. It enables the planning of an optimal trajectory for the AFP head while adhering to the robot motion constraints. However, deviations may occur between the AFP head path and the desired fiber path, and the orientation of the roller in the AFP head could fail to maintain perpendicular to the mandrel surface. For satisfying the AFP geometric constraints, a vision-based trajectory generation strategy is developed for the parallel robot utilizing C-Track 780. Based

on the visual measurement results, the desired parallel robot trajectory can be determined according to the desired trajectory of the start point on the given fiber path.

(3) PBVS-based adaptive sliding mode control of the 6-RSS parallel robot

The trajectory tracking control of the 6-RSS parallel robot in Cartesian space is studied in this project. A position-based visual servoing approach is proposed for the parallel robot using adaptive sliding mode control. The C-Track 780 is employed to measure the end-effector pose of the robot in real time, which serves as the feedback signal to the controller. The adaptive Kalman filter is also utilized to deal with uncertain noises in visual measurements to increase the pose estimation accuracy. A sliding mode controller with strong robustness is designed to cope with system uncertainties, and an RBF neural network is incorporated to realize the auto-tuning of the control gains, which can guarantee the desired system performance under varying conditions. The stability of the controller has been validated using Lyapunov theorem.

(4) DRNN-based adaptive sliding mode control for the cooperative robotic system

In this work, with the existence of dynamic uncertainties and external disturbances, a distributed adaptive sliding mode control scheme using deep recurren neural network is proposed for the cooperative robotic system. A model-based sliding mode controller is developed to maintain the strong robustness of the robotic system. A deep recurrent neural network is designed to estimate the lumped uncertainties in the control system. It consists of a feedforward structure through three hidden layers and a feedback loop from the output layer to the input layer, which exhibits more powerful online learning ability and dynamic property than shallow feedforward neural networks. Based on Lyapunov theorem, the adaptation laws of the neural network parameters are derived, and the stability of the controller can be guaranteed.

(5) Simulation and experimental validations

Sufficient simulation and experiment have been conducted to verify the feasibility of

the proposed trajectory planning and tracking control schemes. The advantages of the algorithms are demonstrated through the comparisons with the previously published research work.

To sum up, the trajectory planning and tracking control of the 13-DOF cooperative robotic system for AFP, have been successfully addressed by algorithm development, theoretical analysis, simulation, and hardware experiment. The distributed control structure with event-triggered configuration is able to realize the collaborative control of two robots in real experiment. The leader-follower trajectory planning approach can satisfy the AFP geometric constraints for 0° fiber path on the Y-shape mandrel, while ensures the compliance to the motion constraints of the robots. The PBVS-based adaptive sliding mode control strategy and the DRNN-based adaptive sliding mode control approach contribute to the enhancement of the robustness and trajectory tracking performance of the cooperative robotic system.

7.2 Future Work

Regarding the trajectory planning and control of the 13-DOF cooperative robotic system for AFP, there still exist several interesting and meaningful topics to investigate, as indicated below.

(1) The traditional sliding mode controller is only insensitive to matched uncertainties, which exist in the same channel as that of the control input [147]. It is worth noting that mismatched uncertainties acting via different channels from that of the control input may appear in the mechanical systems. For example, the uncertainties of the payload dynamics are the mismatched ones while involving the actuator dynamics in the system modeling [148]. In the future, more investigation regarding dealing with the payload in the presence of mismatched uncertainties will be conducted in the controller design for the two robots.

(2) In this work, the PBVS-based adaptive sliding mode control using RBF neural network is only explored for the 6-RSS parallel robot. Since the inverse kinematics solution of a serial robot is not unique, a distinct PBVS control scheme should be developed for the 6-DOF serial robot. Besides, to further test the effectiveness of the proposed control scheme, simulation and experiment using PBVS and DRNN-based adaptive sliding mode controller are worth conducting for the two robots.

(3) Although the employment of neural networks in nonlinear controller development can surely enhance the system robustness to dynamic uncertainties and external disturbances and guarantee the desired tracking performance, it also leads to higher RMS values of the control signals, which makes the controllers more energetically expensive in real applications. More efforts need to be made to decrease the computation cost without compromise on system performance.

(4) In the near future, the experiment of placing the real fiber on the Y-shape mandrel will be implemented. As mentioned in [18], the fiber path with the angles 45° and 90° have also been planned on the Y-shape mandrel. The leader-follower trajectory planning of the cooperative robotic system can be explored in terms of different fiber path angles.

(5) The leader-follower trajectory planning with the parallel robot as the leader and the serial robot as the follower is a potential solution to be studied. Additionally, the trajectory planning for the parallel robot based on optimal control may be considered.

(6) Reinforcement learning can improve the self-learning ability of the robots by enabling the robots to autonomously discover an optimal behavior through the interactions with the environment [149]. The combination of reinforcement learning with sliding mode control or/and deep neural network for robot control can be found in [150–152]. In the future, we will try to integrate the three control techniques for the control of the 13-DOF cooperative robotic system.

(7) Comparisons between distributed control, centralized control and decentralized control using the DRNN-based adaptive sliding mode controller can be investigated on the 13-DOF cooperative robotic system.

(8) The proposed trajectory planning and tracking control strategy can provide inspirations for various cooperative manipulation tasks and the control of different kinds of robots. We will try to implement it on the mobile robots, UAVs and other types of serial robots in our lab for improving the system performance.

Bibliography

[1] L. Zhang, X. Wang, J. Pei, and Y. Zhou, "Review of automated fibre placement and its prospects for advanced composites," *Journal of Materials Science*, vol. 55, no. 17, pp. 7121–7155, 2020.

[2] H.-J. L. Dirk, C. Ward, and K. D. Potter, "The engineering aspects of automated prepreg layup: History, present and future," *Composites Part B: Engineering*, vol. 43, no. 3, pp. 997–1009, 2012.

[3] S. K. Flynn, "2024 State of the Industry Report," 2024. [Online]. Available: https://compositesmanufacturingmagazine.com/2024/02/2024-state-of-the-industry-report-2/

[4] M. S. Sarfraz, H. Hong, and S. S. Kim, "Recent developments in the manufacturing technologies of composite components and their cost-effectiveness in the automotive industry: A review study," *Composite Structures*, vol. 266, p. 113864, 2021.

[5] "Coriolis C1." [Online]. Available: https://www.coriolis-composites.com/fiber-placement-machines/coriolis-c1/

[6] "Electroimpact AFP 4.0 machine." [Online]. Available: https://www.electroimpact.com/Products/composites-manufacturing/automated-fiber-placement-industry-4

[7] "Mikrosam Discovery." [Online]. Available: https://mikrosam.com/product/automated-fiber-and-tape-placement/

[8] "Scalable Composite Robotic Additive Manufacturing Carbon/Carbon." [Online]. Available: https://www.northropgrumman.com/what-we-do/advanced-weapons/scram-c-c

[9] X. Zhang, "Modeling and control of the cooperative automated fiber placement system," Ph.D. dissertation, Concordia University, 2017.

[10] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons, 2020.

[11] J.-P. Merlet, *Parallel robots*, vol. 128. Springer Science & Business Media, 2006.

[12] P. Li, X. Zhang, W. Xie, and S. V. Hoa, "Operation of the collaborative composite manufacturing (CCM) system," *Journal of Visualized Experiments*, no. 152, p. e59969, 2019.

[13] Y. Zhang, S. Li, and L. Liao, "Consensus of high-order discrete-time multiagent systems with switching topology," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 721–730, 2021.

[14] X. Zhang, W. Xie, and S. V. Hoa, "Semi-offline trajectory synchronized algorithm of the cooperative automated fiber placement system," *Robotics and Computer-Integrated Manufacturing*, vol. 51, pp. 53–62, 2018.

[15] N. Zhu, W.-F. Xie, and H. Shen, "Trajectory planning of cooperative robotic system for automated fiber placement in a leader-follower formation," *International Journal of Advanced Manufacturing Technology*, vol. 130, no. 1, pp. 575–588, 2024.

[16] N. Zhu, W.-F. Xie, and H. Shen, "Position-based visual servoing of a 6-RSS parallel robot using adaptive sliding mode control," *ISA Transactions*, vol. 144, pp. 398–408, 2024.

[17] T. Shu, S. Gharaaty, W. Xie, A. Joubair, and I. A. Bonev, "Dynamic path tracking of industrial robots with high accuracy using photogrammetry sensor," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1159–1170, 2018.

[18] C. Hély, L. Birglen, and W.-F. Xie, "Feasibility study of robotic fibre placement on intersecting multi-axial revolution surfaces," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 73–79, 2017.

[19] N. Zhu, W.-F. Xie, and H. Shen, "A leader-follower trajectory planning approach for cooperative robotic system in automated fiber placement," *IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1829–1834, 2023.

[20] C. Heyer, "Human-robot interaction and future industrial robotics applications," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4749–4754, 2010.

[21] X. Yin, L. Pan, and S. Cai, "Robust adaptive fuzzy sliding mode trajectory tracking control for serial robotic manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 72, p. 101884, 2021.

[22] A. Cirillo, P. Cirillo, G. De Maria, A. Marino, C. Natale, and S. Pirozzi, "Optimal custom design of both symmetric and unsymmetrical hexapod robots for aeronautics applications," *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 1–16, 2017.

[23] T. K. Mamidi and S. Bandyopadhyay, "A computational framework for the dynamic analyses of cable-driven parallel robots with feed and retrieval of cables," *Mechanism and Machine Theory*, vol. 186, p. 105338, 2023.

[24] C. Yang, J. Zhao, L. Li, and S. K. Agrawal, "Design and implementation of a novel

modal space active force control concept for spatial multi-DOF parallel robotic manipulators actuated by electrical actuators," *ISA Transactions*, vol. 72, pp. 273–286, 2018.

[25] Q. He, C. Zeng, Z. Gao, and Z. Wu, "Analysis and design of the Stewart platform-based parallel support bumper for inertially stabilized platforms," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 4203–4215, 2020.

[26] R. Wang, S. Li, and Y. Li, "A suspended cable-driven parallel robot with articulated reconfigurable moving platform for schönflies motions," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5173–5184, 2022.

[27] S. Fan, S. Fan, W. Lan, and G. Song, "A new approach to enhance the stiffness of heavy-load parallel robots by means of the component selection," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101834, 2020.

[28] P. Li, T. Shu, W.-F. Xie, and W. Tian, "Dynamic visual servoing of a 6-RSS parallel robot based on optical CMM," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 2, p. 40, 2021.

[29] D. Sun and J. K. Mills, "Adaptive synchronized control for coordination of multirobot assembly tasks," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 498–510, 2002.

[30] A. Zhai, J. Wang, H. Zhang, G. Lu, and H. Li, "Adaptive robust synchronized control for cooperative robotic manipulators with uncertain base coordinate system," *ISA Transactions*, vol. 126, pp. 134–143, 2022.

[31] H.-T. Zhang, H. Xu, B. Xu, Y. Wu, J. Huang, and Q.-L. Han, "Adaptive learning-based distributed control of cooperative robot arm manipulation for unknown objects," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 2, pp. 1298–1307, 2023.

[32] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[33] H. Dou and S. Wang, "Robust adaptive motion/force control for motion synchronization of multiple uncertain two-link manipulators," *Mechanism and Machine Theory*, vol. 67, pp. 77–93, 2013.

[34] X. Jin, "Formation-based decentralized iterative learning cooperative impedance control for a team of robot manipulators," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 2, pp. 872–881, 2023.

[35] R. Cui and W. Yan, "Mutual synchronization of multiple robot manipulators with unknown dynamics," *Journal of Intelligent & Robotic Systems*, vol. 68, pp. 105–119, 2012.

[36] Y. Ren, S. Sosnowski, and S. Hirche, "Fully distributed cooperation for networked uncertain mobile manipulators," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 984–1003, 2020.

[37] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 87–97, 2014.

[38] P. B. g. Dohmann and S. Hirche, "Distributed control for cooperative manipulation

with event-triggered communication," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1038–1052, 2020.

[39] S. Welikala and C. G. Cassandras, "Event-driven receding horizon control for distributed persistent monitoring in network systems," *Automatica*, vol. 127, p. 109519, 2021.

[40] W. S. Cortez, C. K. Verginis, and D. V. Dimarogonas, "A distributed, event-triggered, adaptive controller for cooperative manipulation with rolling contacts," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 3120–3133, 2023.

[41] M. Qian, C. Sun, B. Jiang, R. Wang, and J. Shi, "Event-based adaptive fault tolerant control and collision avoidance of wheel mobile robots with communication limits," *IEEE Transactions on Industrial Electronics*, pp. 1–10, 2024.

[42] X. Zeng, Y. Yang, J. Zhao, and J. Li, "Fully distributed event-triggered control for multi-robot systems based on modal space framework," *Nonlinear Dynamics*, pp. 1–14, 2024.

[43] A. Gasparetto and V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 415–426, 2008.

[44] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[45] H. Liu, X. Lai, and W. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 2, pp. 309–317, 2013.

[46] J. Zhao, G. Yang, H. Shi, S. Chen, C.-Y. Chen, and C. Zhang, "Virtual passive-joint space based time-optimal trajectory planning for a 4-DOF parallel manipulator," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5039–5046, 2023.

[47] E. K. Xidias, "Time-optimal trajectory planning for hyper-redundant manipulators in 3D workspaces," *Robotics and Computer-Integrated Manufacturing*, vol. 50, pp. 286–298, 2018.

[48] C. Ji, Z. Zhang, G. Cheng, M. Kong, and R. Li, "A convex optimization method to time-optimal trajectory planning with jerk constraint for industrial robotic manipulators," *IEEE Transactions on Automation Science and Engineering*, pp. 1–18, 2023.

[49] B. Nadir, O. Mohammed, N. Minh-Tuan, and S. Abderrezak, "Optimal trajectory generation method to find a smooth robot joint trajectory based on multiquadric radial basis functions," *International Journal of Advanced Manufacturing Technology*, vol. 120, no. 1, pp. 297–312, 2022.

[50] J. Huang, P. Hu, K. Wu, and M. Zeng, "Optimal time-jerk trajectory planning for industrial robots," *Mechanism and Machine Theory*, vol. 121, pp. 530–544, 2018.

[51] Y. Wen and P. Pagilla, "Path-constrained and collision-free optimal trajectory planning for robot manipulators," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 763–774, 2023.

[52] J. Hu and G. Feng, "Distributed tracking control of leader–follower multi-agent systems under noisy measurement," *Automatica*, vol. 46, no. 8, pp. 1382–1387, 2010.

[53] Z. Meng, D. V. Dimarogonas, and K. H. Johansson, "Leader–follower coordinated tracking of multiple heterogeneous lagrange systems using continuous control," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 739–745, 2014.

[54] Y. Liu and L. Li, "Adaptive leader-follower consensus control of multiple flexible manipulators with actuator failures and parameter uncertainties," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 4, pp. 1020–1031, 2023.

[55] E. Nuño, D. Valle, I. Sarras, and L. Basañez, "Leader–follower and leaderless consensus in networks of flexible-joint manipulators," *European Journal of Control*, vol. 20, no. 5, pp. 249–258, 2014.

[56] O. O. Obadina, M. A. Thaha, Z. Mohamed, and M. H. Shaheed, "Grey-box modelling and fuzzy logic control of a leader–follower robot manipulator system: A hybrid grey wolf–whale optimisation approach," *ISA transactions*, vol. 129, pp. 572–593, 2022.

[57] D. Zhao, Q. Zhu, N. Li, and S. Li, "Synchronized control with neuro-agents for leader–follower based multiple robotic manipulators," *Neurocomputing*, vol. 124, pp. 149–161, 2014.

[58] S. Islam and X. P. Liu, "Robust sliding mode control for robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2444–2453, 2011.

[59] J. Liu, *Radial Basis Function (RBF) neural network control for mechanical systems: design, analysis and Matlab simulation*. Springer Science & Business Media, 2013.

[60] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[61] A. Perrusquía and W. Yu, "Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview," *Neurocomputing*, vol. 438, pp. 145–154, 2021.

[62] D. Nicolis, F. Allevi, and P. Rocco, "Operational space model predictive sliding mode control for redundant manipulators," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1348–1355, 2020.

[63] B. Ahmadi, W.-F. Xie, and E. Zakeri, "Robust cascade vision/force control of industrial robots utilizing continuous integral sliding-mode control method," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 524–536, 2022.

[64] Y. Wang, Z. Zhang, C. Li, and M. Buss, "Adaptive incremental sliding mode control for a robot manipulator," *Mechatronics*, vol. 82, p. 102717, 2022.

[65] H. Jia, W. Shang, F. Xie, B. Zhang, and S. Cong, "Second-order sliding-mode-based synchronization control of cable-driven parallel robots," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 1, pp. 383–394, 2020.

[66] D. Cruz-Ortiz, I. Chairez, and A. Poznyak, "Non-singular terminal sliding-mode control for a manipulator robot using a barrier Lyapunov function," *ISA Transactions*, vol. 121, pp. 268–283, 2022.

[67] C. Yang, D. Huang, W. He, and L. Cheng, "Neural control of robot manipulators with trajectory tracking constraints and input saturation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 4231–4242, 2021.

[68] G. Peng, C. P. Chen, and C. Yang, "Neural networks enhanced optimal admittance control of robot–environment interaction using reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4551–4561, 2022.

[69] D. Blanck-Kahan, G. Ortiz-Cervantes, V. Martínez-Gama, H. Cervantes-Culebro, J. E. Chong-Quero, and C. A. Cruz-Villar, "Neural-optimal tuning of a controller for a parallel robot," *Expert Systems with Applications*, vol. 236, p. 121184, 2024.

[70] H. Jabbari Asl and F. Janabi-Sharifi, "Adaptive neural network control of cable-driven parallel robots with input saturation," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 252–260, 2017.

[71] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. De Momi, "Deep neural network approach in robot tool dynamics identification for bilateral teleoperation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2943–2949, 2020.

[72] Z. Xu, S. Li, X. Zhou, S. Zhou, T. Cheng, and Y. Guan, "Dynamic neural networks for motion-force control of redundant manipulators: An optimization perspective," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 1525–1536, 2021.

[73] H. Su, Y. Hu, H. R. Karimi, A. Knoll, G. Ferrigno, and E. De Momi, "Improved recurrent neural network-based manipulator control with remote center of motion constraints: Experimental results," *Neural networks*, vol. 131, pp. 291–299, 2020.

[74] C. Liu, G. Wen, Z. Zhao, and R. Sedaghati, "Neural-network-based sliding-mode control of an uncertain robot using dynamic model approximated switching gain," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2339–2346, 2021.

[75] X. Zhao, Z. Liu, B. Jiang, and C. Gao, "Switched controller design for robotic manipulator via neural network-based sliding mode approach," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 2, pp. 561–565, 2023.

[76] V.-T. Nguyen, C.-Y. Lin, S.-F. Su, and Q.-V. Tran, "Adaptive chattering free neural network based sliding mode control for trajectory tracking of redundant parallel manipulators," *Asian Journal of Control*, vol. 21, no. 2, pp. 908–923, 2019.

[77] Y. Zhang, L. Kong, S. Zhang, X. Yu, and Y. Liu, "Improved sliding mode control for

a robotic manipulator with input deadzone and deferred constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 12, pp. 7814–7826, 2023.

[78] M. Zhou, Y. Feng, C. Xue, and F. Han, "Deep convolutional neural network based fractional-order terminal sliding-mode control for robotic manipulators," *Neurocomputing*, vol. 416, pp. 143–151, 2020.

[79] X. Zhang, Y. Huang, Y. Rong, G. Li, H. Wang, and C. Liu, "Recurrent neural network based optimal integral sliding mode tracking control for four-wheel independently driven robots," *IET Control Theory & Applications*, vol. 15, no. 10, pp. 1346–1363, 2021.

[80] K. Zhang, M. Yan, T. Huang, J. Zheng, and Z. Li, "3D reconstruction of complex spatial weld seam for autonomous welding by laser structured light scanning," *Journal of Manufacturing Processes*, vol. 39, pp. 200–207, 2019.

[81] A. Rout, B. Deepak, B. B. Biswal, and G. B. Mahanta, "Weld seam detection, finding, and setting of process parameters for varying weld gap by the utilization of laser and vision sensor in robotic arc welding," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 1, pp. 622–632, 2022.

[82] B. Li, Y. Li, W. Tian, and W. Liao, "Pose accuracy improvement in robotic machining by visually-guided method and experimental investigation," *Robotics and Autonomous Systems*, vol. 164, p. 104416, 2023.

[83] D. K. Bilal, M. Unel, L. T. Tunc, and B. Gonul, "Development of a vision based pose estimation system for robotic machining and improving its accuracy using LSTM neural networks and sparse regression," *Robotics and Computer-Integrated Manufacturing*, vol. 74, p. 102262, 2022.

[84] E. G. Ribeiro, R. Q. Mendes, M. H. Terra, and V. Grassi, "Second-order position-based visual servoing of a robot manipulator," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 207–214, 2024.

[85] T. Dallej, M. Gouttefarde, N. Andreff, P.-E. Hervé, and P. Martinet, "Modeling and vision-based control of large-dimension cable-driven parallel robots using a multiple-camera setup," *Mechatronics*, vol. 61, pp. 20–36, 2019.

[86] P. Yu, N. Tan, and M. Mao, "Position-based visual servo control of dual robotic arms with unknown kinematic models: A cerebellum-inspired approach," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 4, pp. 2328–2339, 2023.

[87] A. Ghasemi, P. Li, and W.-F. Xie, "Adaptive switch image-based visual servoing for industrial robots," *International Journal of Control, Automation and Systems*, vol. 18, no. 5, pp. 1324–1334, 2020.

[88] Z. Li, B. Lai, and Y. Pan, "Image-based composite learning robot visual servoing with an uncalibrated eye-to-hand camera," *IEEE/ASME Transactions on Mechatronics*, pp. 1–11, 2024.

[89] S. Chaudhuri, R. Saha, A. Chatterjee, S. Mookherjee, and D. Sanyal, "Development of a motion sensing system based on visual servoing of an eye-in-hand electrohydraulic parallel manipulator," *IEEE Sensors Journal*, vol. 20, no. 14, pp. 8108–8116, 2020.

[90] Z. Zake, F. Chaumette, N. Pedemonte, and S. Caro, "Robust $2^1/_2$D visual servoing of a cable-driven parallel robot thanks to trajectory tracking," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 660–667, 2020.

[91] Z. Zake, F. Chaumette, N. Pedemonte, and S. Caro, "Visual servoing of cable-driven parallel robots with tension management," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6861–6867, 2021.

[92] K. Jo and D. Chwa, "Robust hybrid visual servoing of omnidirectional mobile manipulator with kinematic uncertainties using a single camera," *IEEE Transactions on Cybernetics*, vol. 54, no. 5, pp. 2824–2837, 2024.

[93] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, "Comparison of basic visual servoing methods," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967–983, 2011.

[94] S. Chen, "Kalman filter for robot vision: a survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012.

[95] J. Wang and W. J. Wilson, "3D relative position and orientation estimation using Kalman filter for robot control," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2638–2639, 1992.

[96] F. Janabi-Sharifi and M. Marey, "A Kalman-filter-based method for pose estimation in visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 939–947, 2010.

[97] A. Saboukhi, "Designing and implementing a small-size automated fiber placement (AFP) head capable of depositing thermoset layers on V-shape structures," Master's thesis, Concordia University, 2023.

[98] J.-C. Yuan, "A general photogrammetric method for determining object position and orientation," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 129–142, 1989.

[99] G. Dong and Z. H. Zhu, "Autonomous robotic capture of non-cooperative target by adaptive extended Kalman filter based visual servo," *Acta Astronautica*, vol. 122, pp. 209–218, 2016.

[100] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[101] D. Pardo, L. Möller, M. Neunert, A. W. Winkler, and J. Buchli, "Evaluating direct transcription and nonlinear optimization methods for robot motion planning," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 946–953, 2016.

[102] M. P. Kelly, "Transcription methods for trajectory optimization: a beginners tutorial," *arXiv preprint arXiv:1707.00284*, 2017.

[103] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[104] R. Zhang, Y. Wang, W. Xie, T. Shu, H. Tan, and Y. Jiang, "Coordination control of the automated fiber placement system using photogrammetry-based leader–follower approach," *Control Engineering Practice*, vol. 141, p. 105691, 2023.

[105] Y. Zhao, H.-C. Lin, and M. Tomizuka, "Efficient trajectory optimization for robot motion planning," *15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 260–265, 2018.

[106] X. Zhang, W.-F. Xie, and S. V. Hoa, "Modeling and workspace analysis of collaborative advanced fiber placement machine," *ASME International Mechanical Engineering Congress and Exposition*, vol. 46476, p. V04AT04A032, 2014.

[107] R. Zeng, S. Dai, W. Xie, and X. Zhang, "Determination of the proper motion range of the rotary actuators of 6-RSS parallel robot," *CCToMM Symposium on Mechanisms, Machines, and Mechatronics*, pp. 28–29, 2015.

[108] T. A. Driscoll, N. Hale, and L. N. Trefethen, "Chebfun guide," 2014.

[109] J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange interpolation," *SIAM review*, vol. 46, no. 3, pp. 501–517, 2004.

[110] J. Waldvogel, "Fast construction of the Fejér and Clenshaw–Curtis quadrature rules," *BIT Numerical Mathematics*, vol. 46, pp. 195–202, 2006.

[111] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: a survey," *Journal of Machine Learning Research*, vol. 18, no. 153, pp. 1–43, 2018.

[112] A. S. Nemirovski and M. J. Todd, "Interior-point methods for optimization," *Acta Numerica*, vol. 17, pp. 191–234, 2008.

[113] Y. Zhao, "Intelligent control and planning for industrial robots," Ph.D. dissertation, University of California, Berkeley, 2018.

[114] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[115] H. Abdellatif and B. Heimann, "Advanced model-based control of a 6-DOF hexapod robot: A case study," *IEEE/ASME Transactions On Mechatronics*, vol. 15, no. 2, pp. 269–279, 2009.

[116] M. Raghavan, "The Stewart platform of general geometry has 40 configurations," *Journal of Mechanical Design*, vol. 115, no. 2, pp. 277–282, 1993.

[117] N. Andreff and P. Martinet, "Unifying kinematic modeling, identification, and control of a Gough–Stewart parallel robot into a vision-based framework," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1077–1086, 2006.

[118] F. Paccot, N. Andreff, and P. Martinet, "A review on the dynamic control of parallel kinematic machines: Theory and experiments," *International Journal of Robotics Research*, vol. 28, no. 3, pp. 395–416, 2009.

[119] P. Li, "Visual calibration, identification and control of 6-RSS parallel robots," Ph.D. dissertation, Concordia University, 2020.

[120] H. Navvabi and A. H. Markazi, "Hybrid position/force control of Stewart manipulator using extended adaptive fuzzy sliding mode controller (E-AFSMC)," *ISA Transactions*, vol. 88, pp. 280–295, 2019.

[121] R. Kumar P., A. Chalanga, and B. Bandyopadhyay, "Smooth integral sliding mode controller for the position control of Stewart platform," *ISA Transactions*, vol. 58, pp. 543–551, 2015.

[122] Y. Zhou, J. She, F. Wang, and M. Iwasaki, "Disturbance rejection for Stewart platform based on integration of equivalent-input-disturbance and sliding-mode control methods," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 4, pp. 2364–2374, 2023.

[123] T. Huang, L. Shan, J. Li, Z. Yu, and C. Liu, "Closed-loop RBF-PID control method for position and attitude control of Stewart platform," *Chinese Control Conference (CCC)*, pp. 3096–3101, 2019.

[124] M. Taghizadeh and M. Javad Yarmohammadi, "Development of a self-tuning PID controller on hydraulically actuated Stewart platform stabilizer with base excitation," *International Journal of Control, Automation and Systems*, vol. 16, pp. 2990–2999, 2018.

[125] N. N. Son, C. Van Kien, and H. P. H. Anh, "A novel adaptive feed-forward-PID controller of a SCARA parallel robot using pneumatic artificial muscle actuator based

on neural network and modified differential evolution algorithm," *Robotics and Autonomous Systems*, vol. 96, pp. 65–80, 2017.

[126] N. Zhu, W. Xie, and H. Shen, "Adaptive sliding mode control with RBF neural network-based tuning method for parallel robot," *48th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pp. 1–6, 2022.

[127] M. Ertugrul and O. Kaynak, "Neuro sliding mode control of robotic manipulators," *Mechatronics*, vol. 10, no. 1-2, pp. 239–263, 2000.

[128] M. Huang, X. Huang, X. Tu, Z. Li, and Y. Wen, "An online gain tuning proxy-based sliding mode control using neural network for a gait training robotic orthosis," *Cluster Computing*, vol. 19, pp. 1987–2000, 2016.

[129] H.-V.-A. Truong, D.-T. Tran, and K. K. Ahn, "A neural network based sliding mode control for tracking performance with parameters variation of a 3-DOF manipulator," *Applied Sciences*, vol. 9, no. 10, p. 2023, 2019.

[130] Z. Yang, J. Wu, and J. Mei, "Motor-mechanism dynamic model based neural network optimized computed torque control of a high speed parallel manipulator," *Mechatronics*, vol. 17, no. 7, pp. 381–390, 2007.

[131] S. Rahimi, H. Jalali, M. R. H. Yazdi, A. Kalhor, and M. T. Masouleh, "Design and practical implementation of a neural network self-tuned inverse dynamic controller for a 3-DoF Delta parallel robot based on arc length function for smooth trajectory tracking," *Mechatronics*, vol. 84, p. 102772, 2022.

[132] T. D. Le, H.-J. Kang, Y.-S. Suh, and Y.-S. Ro, "An online self-gain tuning method using neural networks for nonlinear PD computed torque controller of a 2-dof parallel manipulator," *Neurocomputing*, vol. 116, pp. 53–61, 2013.

[133] B. Achili, B. Daachi, Y. Amirat, and A. Ali-Cherif, "A robust adaptive control of a parallel robot," *International Journal of control*, vol. 83, no. 10, pp. 2107–2119, 2010.

[134] T. D. Le, H.-J. Kang, and Y.-S. Suh, "Chattering-free neuro-sliding mode control of 2-DOF planar parallel manipulators," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 22, 2013.

[135] A. T. Vo and H.-J. Kang, "Neural integral non-singular fast terminal synchronous sliding mode control for uncertain 3-DOF parallel robotic manipulators," *IEEE Access*, vol. 8, pp. 65 383–65 394, 2020.

[136] T. Dallej, N. Andreff, Y. Mezouar, and P. Martinet, "3D pose visual servoing relieves parallel robot control from joint sensing," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4291–4296, 2006.

[137] S. P. Bhat and D. S. Bernstein, "Finite-time stability of continuous autonomous systems," *SIAM Journal on Control and Optimization*, vol. 38, no. 3, pp. 751–766, 2000.

[138] J. Kang, W. Meng, A. Abraham, and H. Liu, "An adaptive PID neural network for complex nonlinear system control," *Neurocomputing*, vol. 135, pp. 79–85, 2014.

[139] S. Cong and Y. Liang, "PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3872–3879, 2009.

[140] O. Khatib, P. Thaulad, T. Yoshikawa, and J. Park, "Torque-position transformer for task control of position controlled robots," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1729–1734, 2008.

[141] N. Adhikary and C. Mahanta, "Inverse dynamics based robust control method for position commanded servo actuators in robot manipulators," *Control Engineering Practice*, vol. 66, pp. 146–155, 2017.

[142] T. N. Truong, A. T. Vo, and H.-J. Kang, "Neural network-based sliding mode controllers applied to robot manipulators: A review," *Neurocomputing*, vol. 562, p. 126896, 2023.

[143] Y. Chu, J. Fei, and S. Hou, "Adaptive global sliding-mode control for dynamic systems using double hidden layer recurrent neural network structure," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1297–1309, 2020.

[144] J. Fei, Y. Chen, L. Liu, and Y. Fang, "Fuzzy multiple hidden layer recurrent neural control of nonlinear system using terminal sliding-mode controller," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9519–9534, 2022.

[145] V. T. Yen, W. Y. Nan, and P. Van Cuong, "Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators," *Neural Computing and Applications*, vol. 31, no. 11, pp. 6945–6958, 2019.

[146] J. Hu, D. Zhang, Z.-G. Wu, and H. Li, "Neural network-based adaptive second-order sliding mode control for uncertain manipulator systems with input saturation," *ISA Transactions*, vol. 136, pp. 126–138, 2023.

[147] J. Yang, S. Li, and X. Yu, "Sliding-mode control for systems with mismatched uncertainties via a disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 1, pp. 160–169, 2013.

[148] W. Deng and J. Yao, "Asymptotic tracking control of mechanical servosystems with

mismatched uncertainties," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 2204–2214, 2021.

[149] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[150] N. Sacchi, G. P. Incremona, and A. Ferrara, "Sliding mode based fault diagnosis with deep reinforcement learning add-ons for intrinsically redundant manipulators," *International Journal of Robust and Nonlinear Control*, vol. 33, no. 15, pp. 9109–9127, 2023.

[151] X. Wang, D. Wang, M. Du, K. Song, Y. Ni, and Y. Li, "A two-layer trajectory tracking control scheme of manipulator based on ELM-SMC for autonomous robotic vehicle," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2023.

[152] Y. Lu, C. Wu, W. Yao, G. Sun, J. Liu, and L. Wu, "Deep reinforcement learning control of fully-constrained cable-driven parallel robots," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 7, pp. 7194–7204, 2023.