DEVELOPMENT OF LEARNING CONTROL STRATEGIES FOR A CABLE-DRIVEN DEVICE ASSISTING A HUMAN JOINT

by

Hao Xiong

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Engineering Technology

West Lafayette, Indiana

December 2019

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Xiumin Diao, Chair
School of Engineering Technology
Dr. Duane Dunlap
School of Engineering Technology
Dr. Daniel Leon-Salas
School of Engineering Technology
Dr. Suranjan Panigrahi
School of Engineering Technology
Dr. Haiyan Zhang
School of Engineering Technology

Approved by:

Dr. Kathryne A Newton
Head of the Graduate Program

ACKNOWLEDGMENTS

First of all, I have to extend my heartfelt gratitude to my advisor, Dr. Xiumin Diao, for his extraordinary advising, strong support, and consistent encouragement. He brought me a lifelong impression of his intelligence and diligence.

Special appreciation should be given to my Committee members, Professors Duane D. Dunlap, Daniel Leon-Salas, Suranjan Panigrahi, and Haiyan Zhang from Purdue University. Their precious and insightful guidance is sincerely appreciated. Thanks are also due to Dr. Lin Zhang, Shengchao Li, Jin Hu, Yu Mei, and Tianqi Ma from Purdue University and Zhongyuan Liu from Shanghai Jiaotong University for our marvelous cooperation. Moreover, I want to express my gratitude to all the staff members from Purdue University for their selfless help. Last but not least, I would like to thank my family members for their continued support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADLs	Activities of Daily Living
ANN	Artificial Neural Network
CDAD	Cable-Driven Assistive Device
CDPR	Cable-Driven Parallel Robot
DDPG	Deep Deterministic Policy Gradient
DNN	Deep Neural Network
DOF	Degree of Freedom
DPG	Deterministic Policy Gradient
DQN	Deep Q Network
DRL	Deep Reinforcement Learning
EWII	Entire Workspace Isotropy Index
FNN	Feedforward Neural Network
GPS	Guided Policy Search
IAC	Isotropy Analysis Center
IMU	Inertial Measurement Unit
LQR	Linear Quadratic Regulator
MDGPS	Mirror Descent Guided Policy Search
MDP	Markov Decision Process
NAF	Normalized Advantage Function
PID	Proportional - Integral - Derivative
PMA	Pneumatic Muscle Actuator
PPO	Proximal Policy Optimization
RAD	Robotic Assistive Device

RL	Reinforcement Learning
RNN	Recurrent Neural Network
RWII	Rotational Workspace Isotropy Index
SEA	Serial Elastic Actuator
SSV	Structured Singular Value
TRPO	Trust Region Policy Optimization
TWII	Translational Workspace Isotropy Index
UUB	Uniformly Ultimately Bounded

ABSTRACT

Author: Xiong, Hao. PhD
Institution: Purdue University
Degree Received: December 2019
Title: Development of Learning Control Strategies for a Cable-Driven Device Assisting a Human Joint
Committee Chair: Xiumin Diao

There are millions of individuals in the world who currently experience limited mobility as a result of aging, stroke, injuries to the brain or spinal cord, and certain neurological diseases. Robotic Assistive Devices (RADs) have shown superiority in helping people with limited mobility by providing physical movement assistance. However, RADs currently existing on the market for people with limited mobility are still far from intelligent.

Learning control strategies are developed in this study to make a Cable-Driven Assistive Device (CDAD) intelligent in assisting a human joint (e.g., a knee joint, an ankle joint, or a wrist joint). CDADs are a type of RADs designed based on Cable-Driven Parallel Robots (CDPRs). A PID–FNN control strategy and DDPG-based strategies are proposed to allow a CDAD to learn physical human-robot interactions when controlling the pose of the human joint. Both pose-tracking and trajectory-tracking tasks are designed to evaluate the PID–FNN control strategy and the DDPG-based strategies through simulations. Simulations are conducted in the Gazebo simulator using an example CDAD with three degrees of freedom and four cables. Simulation results show that the proposed PID–FNN control strategy and DDPG-based strategies work in controlling a CDAD with proper learning.

CHAPTER 1. INTRODUCTION

For chapter one, the problem that motivates this study is introduced: Robotic Assistive Devices (RADs) used for people with limited mobility are still far from intelligent. The study aims to take advantage of the recent development of artificial intelligence technologies to answer the research question - how learning control strategies can make Cable-Driven Assistive Devices (CDADs) intelligent when assisting a human joint. The significance, scope, assumptions, limitations, and delimitations of this study are introduced as well.

1.1 Problem Statement

The percentage of the elder people is increasing along with the worldwide population aging. The weaknesses of the skeletal muscles trouble Activities of Daily Living (ADLs) of the elder people (Huo, Mohammed, Moreno, & Amirat, 2016). Moreover, there are millions of individuals in the world who currently experience various movement-related disabilities (Babaiasl, Mahdioun, Jaryani, & Yazdani, 2016) frequently as a result of stroke, injuries to the brain or spinal cord, and certain neurological diseases (Maciejasz, Eschweiler, Gerlach-Hahn, Jansen-Troy, & Leonhardt, 2014). Healthcare research has provided evidence for those with movement-related disabilities can restore their impaired movement capabilities by performing repetitive rehabilitation training of their impaired extremities (Barreca, Wolf, Fasoli, & Bohannon, 2003; Dobkin, 2004). Robots have shown superiority in helping people with limited mobility to provide physical movement assistance (Barreca et al., 2003). However, current RADs for people with limited mobility (e.g., the elderly and people with movement-related disabilities) are still far from intelligent.

1.2 Significance

According to a survey from the United Nations Population Fund, in 2012, people aged 60 and over represent 11.5% of the global population, and the percentage is predicted to 22% by 2050 (UNFPA, 2012). Aging is a particularly prominent problem in regions and countries. For example, according to China's national 6th demographic census in 2010, the proportion of the population aged 65 and over in the total population is 8.9%, indicating that China is the country with the biggest aged

population in the world (Liu, Lin, Lin, & Zheng, 2018). The weakness of the skeletal muscles seriously limits the elderly to achieve their ADLs without intelligent RADs (Huo et al., 2016).

Moreover, stroke is the leading cause of permanent disability in industrialized nations. Over 920,000 Europeans and 700,000 North Americans have a stroke each year, and more than a half survive but with severe impairments (Morales, Badesa, García-Aracil, Sabater, & Pérez-Vidal, 2011). The survivors of stroke can experience paralysis or loss of physical strength making it difficult to perform ADLs (Lo & Xie, 2012). Healthcare research has provided evidence that people with movement-related disabilities can restore their impaired movement capabilities by performing repetitive rehabilitation training of their impaired extremities (Barreca et al., 2003; Dobkin, 2004). However, the traditional manual rehabilitation training processes, without intelligent RADs, are labor-intensive and time-consuming (Fazekas, Horvath, Troznai, & Toth, 2007; Kwakkel, Kollen, & Krebs, 2008; Prange, Jannink, Groothuis-Oudshoorn, Hermens, & Ijzerman, 2009).

1.3 Research Question

Compared to other RADs, CDADs offer several merits, such as low inertia, high payload-to-weight ratio, modularity, simple architecture, and convenient for reconfiguration (Du & Agrawal, 2015; Mayhew, Bachrach, Rymer, & Beer, 2005). Thus, CDADs are suitable for providing movement assistance for people with limited mobility. Since a CDAD is with modularity, simple architecture, and convenient for reconfiguration, the same CDAD is potential to be used in various scenarios in assisting people with limited mobility. However, the control of the CDAD is challenging because of the inconsistent setups, modularity of the CDAD, and reconfiguration of the CDAD. The study focuses on the following question - how to apply learning control strategies to a CDAD assisting a human joint.

1.4 Scope

Learning control strategies are developed for a CDAD in this study. The CDAD is designed to assist a human joint adaptively. Learning control strategies proposed in this study are based on Artificial Neural Networks (ANNs) or Deep Reinforcement Learning (DRL) algorithms. The

interactions between a person and a CDAD is twofold: physical human-robot interaction and cognitive human-robot interaction (Gopura, Bandara, Kiguchi, & Mann, 2016). The CDAD is supposed to learn the physical human-robot interactions, not the cognitive human-robot interactions.

The learning control strategies will be verified via simulations in this study. Since upper-level assistive strategies or rehabilitation strategies (Xiong & Diao, 2019) of CDADs set target pose or target trajectory inevitably, pose-tracking and trajectory-tracking tasks are designed to evaluate the proposed learning control strategies. An example CDAD with four cables will be established in the Gazebo simulator (Koenig & Howard, 2004), a physical simulator, to assist a joint with three Degrees Of Freedom (DOFs). The example CDAD with proposed learning control strategies will learn the physical human-robot interactions and then assist the joint in the Gazebo simulator.

The following research tasks are carried out to answer the research question:

- Review literature on RADs, CDADs, Cable-Driven Parallel Robots (CDPRs), ANNs, and DRL.
- Propose learning control strategies to allow a CDAD to adapt to physical human-robot interaction when assisting a human joint.
- Evaluate the learning control strategies through simulation.

1.5 Assumptions

The following assumptions were inherent to the study:

- A CDAD is designed for a human joint that has up to three DOFs.
- The setup of a CDAD is not consistent in each time.
- The CDAD works in the pose tracking mode and the target pose depends on a higher-level assistive mode (e.g., partially assistive control, triggered passive control, and passive control) (Proietti, Crocher, Roby-Brami, & Jarrassé, 2016).
- The CDAD is driven by torque/force-controlling actuators (e.g., Direct-Current motors).

1.6 Limitation

The following limitation was inherent to this study:

- The CDAD is not applied to clinical trials.

1.7 Delimitations

The study is conducted with the following delimitations:

- Since a human joint is concerned, the optimal design of the configuration of a CDAD for a specific human joint is not discussed in this study.
- Since assistive modes of RADs (e.g., partially assistive control, triggered passive control, and passive control) (Basteris et al., 2014; Cao, Xie, Das, & Zhu, 2014; Jarrassé et al., 2014; Marchal-Crespo & Reinkensmeyer, 2009; Proietti et al., 2016; Yan, Cempini, Oddo, & Vitiello, 2015) can also be applied to CDADs, the discussion of assistive modes of CDADs is not mentioned in this study.

1.8 Dissertation Layout

The dissertation is organized into five chapters. The dissertation starts with an introduction in chapter one. For chapter two, studies related to CDPRs, CDADs, requirements of RADs, pros and cons of CDADs, ANNs, and DRL are reviewed. For chapter three, the methodologies used in the dissertation are introduced. Fundamental studies for CDADs are discussed at first. A PID–FNN control strategy and DDPG-based strategies are proposed for a CDAD. Simulations are designed to verify the proposed learning control strategies. Chapter four presents the findings from the fundamental studies on CDPRs and CDADs and the evaluation results of the proposed learning control strategies in simulations. The dissertation is summarized in chapter five.

CHAPTER 2. REVIEW OF LITERATURE

A review of literature related to CDADs and learning control strategies is necessary to develop learning control strategies for a CDAD that can adaptively assist a human joint. For chapter two, studies of cable-driven parallel robots, cable-driven assistive devices, requirements of robotic assistive devices, pros and cons of cable-driven assistive devices, artificial neural networks, and deep reinforcement learning are reviewed. Sections 2.2, 2.3, and 2.4 of chapter two are revised based on the author's published paper (Xiong & Diao, 2019). The review of literature related to CDADs and learning control strategies comprises searches in the databases of PubMed, IEEE Xplore Digital Library, Science Direct, and Google Scholar using various combinations of the following keywords: cable, wire, rehabilitation, assistance, therapy, training, robot, elastic, pneumatic, neural networks, and deep reinforcement learning.

2.1 Cable-Driven Parallel Robots

Since a CDAD is based on a CDPR in this study, studies related to the CDPR are necessary to be reviewed. The kinematics, dynamics, workspace, stiffness and control aspects of the CDPR are presented in this study.

2.1.1 Kinematics

Based on a CDPR that assists a joint via four cables, the kinematics architecture of the CDPR, as shown in Figure 2.1, is introduced. A_i and B_i are the two attaching points of the *i*th cable on the base and the end-effector, respectively. $\boldsymbol{l}_i \in R^3$ $(i = 1, 2, 3, 4)$ is the vector along the *i*th cable. $\boldsymbol{u}_i$ is the unit vector along the *i*th cable. The magnitude of vector $\boldsymbol{l}_i$, represented by scalar l_i, is the length of the *i*th cable between attaching points A_i and B_i. The positions of the attaching points A_i and B_i are represented by vectors $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ defined in the base frame, respectively. The origin of the base frame F_b is denoted by point *O*, and the origin of the end-effector frame F_e is denoted as point *P*. Both *O* and *P* locate at the rotation center of the end-effector, namely the rotation center of the joint. The attitude of F_e with respect to F_b is described by three Euler angles $\boldsymbol{\Omega} = [\psi, \theta, \phi]^T$ with the ψ-θ-ϕ sequence.

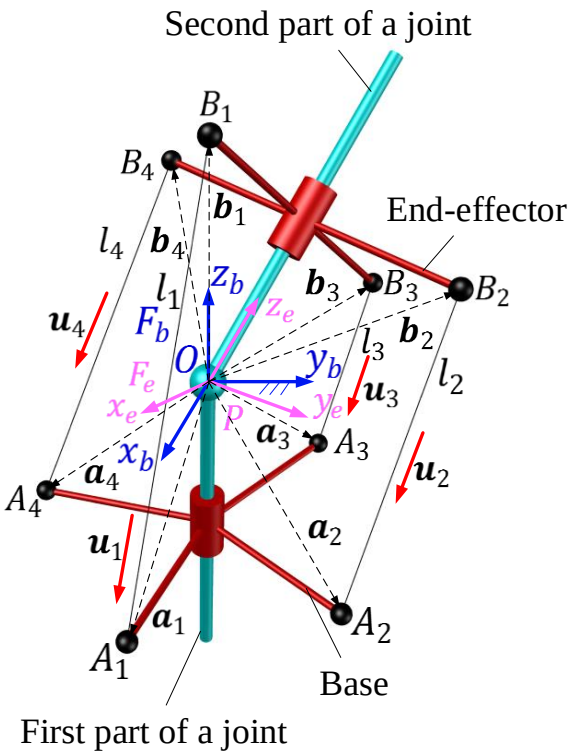


Figure 2.1 Kinematics notations of a joint assisted by a CDPR

Based on the above kinematics notations, the position of the end-effector with respect to the base frame can be described as (Diao & Ma, 2007)

$$\boldsymbol{p} = \boldsymbol{a}_i - \boldsymbol{l}_i - \boldsymbol{b}_i \ \ for\ i = 1,2,3,4 \tag{2.1}$$

from which one has

$$l_i^2 = [\boldsymbol{a}_i - \boldsymbol{p} - \boldsymbol{b}_i]^{\boldsymbol{T}}[\boldsymbol{a}_i - \boldsymbol{p} - \boldsymbol{b}_i] \ \ for\ i = 1,2,3,4 \tag{2.2}$$

Differentiating (2.2) with respect to time, and then organizing the four resulting equations into a matrix form, one obtains

$$\dot{\boldsymbol{l}} = -\boldsymbol{J}\dot{\boldsymbol{X}} \tag{2.3}$$

where

$$\dot{\boldsymbol{l}} = [\dot{l}_1 \quad \dot{l}_2 \quad \dot{l}_3 \quad \dot{l}_4]^T \tag{2.4}$$

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \boldsymbol{u}_3 & \boldsymbol{u}_4 \\ \boldsymbol{b}_1 \times \boldsymbol{u}_1 & \boldsymbol{b}_2 \times \boldsymbol{u}_2 & \boldsymbol{b}_3 \times \boldsymbol{u}_3 & \boldsymbol{b}_4 \times \boldsymbol{u}_4 \end{bmatrix}^T \tag{2.5}$$

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{v}_p \\ \boldsymbol{\omega} \end{bmatrix} = [\dot{x}\ \dot{y}\ \dot{z}\ \dot{\theta}_x\ \dot{\theta}_y\ \dot{\theta}_z]^T \tag{2.6}$$

For the equations (2.1)-(2.6), $\dot{\boldsymbol{l}}$ is the velocity vector in joint space. $\dot{\boldsymbol{X}}$ denotes the twist of the end-effector. $\boldsymbol{v}_p = [\dot{x}\ \dot{y}\ \dot{z}]^T$ and $\boldsymbol{\omega} = \left[\dot{\theta}_x\ \dot{\theta}_y\ \dot{\theta}_z\right]^T$ represent the translational and rotational velocity vectors of F_b with respect to F_e. $\boldsymbol{J}$ is the Jacobian (Diao & Ma, 2009) of the CDPR.

2.1.2 Dynamics

Compared to the inertia of the end-effector, the inertia of cables is usually much smaller and can be ignored (Ottaviano & Castelli, 2010). Then, the motion of the end-effector can be expressed based on Newton-Euler Formulation as (Khosravi & Taghirad, 2014b)

$$\boldsymbol{M}(\boldsymbol{X})\ddot{\boldsymbol{X}} + \boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)\dot{\boldsymbol{X}} + \boldsymbol{G}(\boldsymbol{X}) + \boldsymbol{w}_e = \boldsymbol{w} \tag{2.7}$$

where $\boldsymbol{M}(\boldsymbol{X})$ denotes the mass matrix. $\boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)$ denotes the Coriolis and centripetal matrix. $\boldsymbol{G}(\boldsymbol{X})$ represents the gravity vector. $\boldsymbol{w}_e$ represents the external wrench vector. $\boldsymbol{w}$ is a wrench vector applied on the CDPR via cables. $\boldsymbol{X}$ represents the pose of the CDPR. The mass matrix $\boldsymbol{M}(\boldsymbol{X})$, Coriolis and centripetal matrix $\boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)$, and the gravity matrix $\boldsymbol{G}(\boldsymbol{X})$ have been defined by several scholars (Khosravi & Taghirad, 2014b; Ma & Diao, 2005)

$$\boldsymbol{M}(\boldsymbol{X}) = \begin{bmatrix} m\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_p \end{bmatrix} \tag{2.8}$$

$$\boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)\dot{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{\omega} \times m\boldsymbol{v}_p \\ \boldsymbol{\omega} \times \boldsymbol{I}_p\boldsymbol{\omega} \end{bmatrix} \tag{2.9}$$

$$\boldsymbol{G}(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{F}_g \\ \boldsymbol{w}_g \end{bmatrix} \tag{2.10}$$

where $\boldsymbol{v}_p$ is the translational velocity vector of the center of mass of the end-effector. $\boldsymbol{I}_p$ is the inertia tensor of the end-effector about point O in F_b. m is the mass of the end-effector. $\boldsymbol{I}$ is the identity matrix. $\boldsymbol{0}$ is the zero matrix. $\boldsymbol{F}_g$ is the gravity of the end-effector, and $\boldsymbol{w}_g$ is the moment of the gravity.

If there is no collision between the cables, base, and end-effector, and the Jacobian of the CDPR is certain, the set of cable tensions required to generate a certain wrench on the end-effector can be obtained by solving the inverse dynamics problem (Khosravi & Taghirad, 2014b) as

$$\boldsymbol{w} = -\boldsymbol{J}^T\boldsymbol{\tau} \tag{2.11}$$

where $\boldsymbol{\tau}$ is a cable tension vector. However, if the Jacobian of the CDPR is uncertain, the set of cable tensions required to generate a certain wrench cannot be obtained based on (2.11).

2.1.3 Workspace

The workspace of a CDPR was discussed by researchers in the past decades. Generally speaking, a force-closure workspace of a CDPR is defined as the set of all the poses (i.e., both positions and orientations) that the end-effector can physically reach while the force feasibility condition (i.e., all the cables are in tension) and perhaps additional constraints leading to different types of workspace (Diao & Ma, 2008). The controllable workspace (Verhoeven & Hiller, 2000) is defined as the set of all the poses at which the end-effector can statically balance a specific set of external wrenches with positive cable forces. Barrette and Gosselin introduced the concept of dynamic workspace, defined as the set of all end-effector poses and accelerations at which the end-effector can be balanced with non-negative cable forces (Barrette & Gosselin, 2005). The static workspace (Pusey, Fattah, Agrawal, & Messina, 2004) is defined as the set of all the poses at which the end-effector can be statically balanced with non-negative cable forces.

2.1.4 Stiffness

The stiffness of a CDPR has been discussed by a few researchers. Behzadipour and Sohi (Behzadipour & Sohi, 2007) introduced the antagonistic cable tensions of a CDPR and proposed the concepts of antagonistic stiffness and elastic stiffness. The antagonistic cable tensions can generate antagonistic stiffness in addition to the conventional elastic stiffness of a CDPR. The stability of a CDPR with respect to antagonistic cable tensions was discussed (Behzadipour & Khajepour, 2005). A positive semi-definite antagonistic stiffness matrix is sufficient and necessary to guarantee the corresponding pose of the CDPR is stable. A set of sufficient conditions was derived to achieve the positive semi-definite antagonistic stiffness matrix of a CDPR. Moreover, it was proved (B. Zhang, Shang, Cong, & Liu, 2017) that the sufficient and necessary condition for a CDPR to have a stable pose in the force-closure workspace is that its Jacobian matrix has a full rank and the active stiffness matrix is positive definite. The concept of stiffness ratio was proposed (Azadi, Behzadipour, & Faulkner, 2009) to demonstrate the ratio of the antagonistic stiffness to the stiffness (i.e., the antagonistic stiffness plus the elastic stiffness) of a CDPR. The maximum and the minimum stiffness ratios of the CDPR were investigated when the CDPR is in singular configurations.

2.1.5 Control

The control of CDPRs has also been investigated by researchers. Proportional–Integral–Derivative controllers (PID controllers) were wildly used in the control of CDPRs (Alamdari & Krovi, 2015a; Brackbill, Mao, Agrawal, Annapragada, & Dubey, 2009; W. Chen, Cui, Zhang, & Wang, 2015; Cui, Chen, Jin, & Agrawal, 2017; Feki et al., 2013). The control of CDPRs based on optimal controllers (Jamshidifar, Khosravani, Fidan, & Khajepour, 2018; Meunier, Boulet, & Nahon, 2009) as well as fuzzy controllers (Wang, Tong, Yi, & Li, 2015; Zi, Duan, Du, & Bao, 2008) was also discussed by researchers. Alikhani and Vali proposed a sliding mode controller to control a CDPR (Alikhani & Vali, 2011). Adaptive controllers have a reliable performance in the control of CDPRs (Babaghasabha, Khosravi, & Taghirad, 2015; Kino, Yoshitake, Wada, Tahara, & Tsuda, 2018; Lamaury, Gouttefarde, Chemori, & Herve, 2013). Singular perturbation approach was developed to control CDPRs with elastic cables (Khosravi & Taghirad, 2014a). A passivity-based control method was applied to CDPRs (Caverly & Forbes, 2014). Moreover, computed torque methods showed promising performance in controlling CDPRs (Alp & Agrawal, 2002; Williams, Gallina, & Vadia, 2003). In order to deal with uncertainties of CDPRs, the robustness of controllers was studied (Babaghasabha et al., 2015; Jamshidifar et al., 2018; Khosravi & Taghirad, 2014b; Kino, Yahiro, Takemura, & Morizono, 2007). All the above-mentioned control strategies assume the Jacobians (Diao & Ma, 2009) of the CDPRs can be determined in the control process.

2.2 Current Cable-Driven Assistive Devices

Robotic assistive devices can provide customized, prolonged, intensive, and repetitive movement assistance to people with limited mobility (Niyetkaliyev, Hussain, Ghayesh, & Alici, 2017). RADs from stationary ones to portable ones have become widespread and clinically accepted in the past decade. Cable-driven assistive devices are a type of RADs designed based on CDPRs. One example CDAD for ankle rehabilitation is shown in Figure 2.2. The cuff worn on the shank is the base of the CDPR while the brace on the foot is the end-effector of the CDPR. The end-effector is connected to the base through four cables in this example. Anchor points are points where cables are connected to the end-effector and the base. Because of the benefits of CDADs, the CDADs have attracted the attention of researchers.

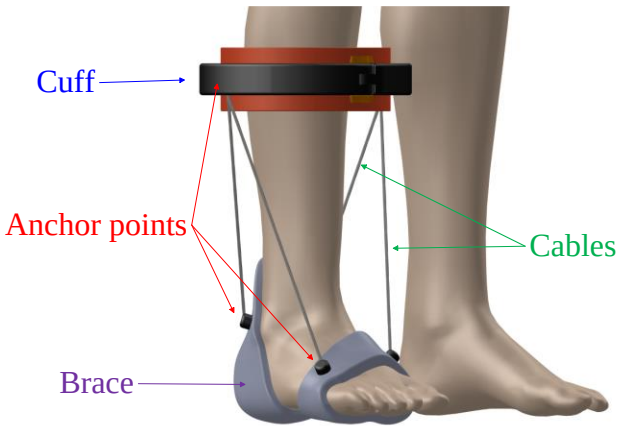


Figure 2.2 One example CDAD for ankle rehabilitation

2.2.1 Cable-Driven Assistive Devices for Upper Limbs and Fingers

CDADs for upper limb have been developed in the past years. Cuffs worn by a person are used on the CDADs. Specifically, CAREX is a five-DOF CDAD with three cuffs (i.e., shoulder cuff, upper arm cuff, and forearm cuff) (Mao & Agrawal, 2012). A seven-DOF CDAD with four cuffs (i.e., shoulder cuff, upper arm cuff, forearm cuff, and hand cuff) and eight cables was proposed (Cui et al., 2017; Cui, Chen, Zhang, & Wang, 2015). A three-DOF cable-driven shoulder exoskeleton with two cuffs (i.e., shoulder cuff and upper arm cuff), which is a simplified version of CAREX, was conceptually developed (Shao, Tang, & Yi, 2014) for shoulder rehabilitation. Moreover, CARR-4 (Z. Li, Chen, Zhang, & Bai, 2017) is a CDAD with six cables and three cuffs (i.e., shoulder cuff, upper arm cuff, and forearm cuff) for the rehabilitation of shoulders and elbows. The CARR-4 can provide three DOFs in shoulder rehabilitation and one DOF in elbow rehabilitation. Moreover, a CDAD consisting of four cables and two cuffs (i.e., forearm cuff and hand cuff) was designed (W. Chen et al., 2015) for wrist rehabilitation. The design of CDADs is based on cuffs (e.g., shoulder cuff, upper arm cuff, forearm cuff, and hand cuff) and a group of cables. The cables connect one or more cuffs and are driven by actuators usually mounted on a fixed platform. The motion of the cuffs and the corresponding segments of a person is controlled by the group of cables simultaneously.

Soft braces are employed in CDADs for upper limb rehabilitation as well. A soft cable-driven one-DOF exoskeleton was developed (Galiana, Hammond, Howe, & Popovic, 2012) and (Kesner,

Jentoft, Hammond, Howe, & Popovic, 2011) for shoulder rehabilitation. The CDAD has one shoulder brace, one elbow brace, two Bowden cables, and two Series Elastic Actuators (SEAs). The abduction and adduction motion of a human shoulder are controlled by both Bowden cables attached to the braces. A Bowden-cable-driven soft exoskeleton was proposed for elbow rehabilitation (Wei, Qu, Wang, Zhang, & Hao, 2018). The CDAD utilizes four cables and strips to assist the flexion and extension of an elbow. Moreover, Auxilio, a CDAD with three cables for shoulder and elbow rehabilitation, was proposed (Gaponov, Popov, Lee, & Ryu, 2017). The Auxilio provides two DOFs in shoulder rehabilitation and one DOF (i.e., flexion) in elbow rehabilitation. The design of CDADs is based on soft braces and a group of cables. Since soft braces, strips, and cables are soft and compact, CDADs are potentially able to be smuggled underneath clothing.

CDADs have been untied to finger rehabilitation by several researchers, such as CADEX (D. H. Kim & Park, 2018) and others (In, Cho, Kim, & Lee, 2011; Jeong, In, & Cho, 2013; Kang, In, & Cho, 2012; Mohamaddan & Komeda, 2010; S. Park, Weber, Bishop, Stein, & Ciocarlie, 2018; Yeow, Baisch, Talbot, & Walsh, 2014). Several CDPRs with two cables are used in CDADs to control one-DOF (i.e., flexion and extension) of fingers. Although various rigid and soft sleeves are utilized, CDADs are free of revolute axes. Thus, the CDADs have adaptability and do not have misalignment problems.

2.2.2 Cable-Driven Assistive Devices for Lower Limbs

Several CDADs for lower limb rehabilitation have been discussed by researchers. Cuffs are used by the CDADs. A cable-driven three-DOF exoskeleton for lower limb rehabilitation, named C-ALEX, was developed (X. Jin, Cui, & Agrawal, 2015). Furthermore, C-ALEX has three cuffs (i.e., waist cuff, thigh cuff, and shank cuff). Ankle prosthesis can be driven by four cables and two cuffs (i.e., shank cuff and foot cuff) for gait training (Dhir, Dallali, Ficanha, Ribeiro, & Rastgaar, 2018). Also, a CDAD driven by four Pneumatic Muscle Actuators (PMAs) was designed for ankle rehabilitation (Jamwal, Xie, & Aw, 2009).

Several CDADs for lower limb rehabilitation have soft braces. “Second Skin” is a one-DOF CDAD proposed (Y. Park, Santos, Galloway, Goldfield, & Wood, 2014) for knee rehabilitation.

"Second Skin" is composed of PMAs and soft sleeves. A series of soft "exosuits" were proposed and developed (Bae et al., 2018; J. Kim et al., 2018; Lee et al., 2018; Wehner et al., 2013) for lower limb assistance and rehabilitation. The CDADs consist of cables (or PMAs) and soft braces, so the CDADs possess the inherent safety.

A bio-inspired CDAD was developed (Nagpal, 2014) for ankle-foot rehabilitation. The design of this CDAD is based on one knee brace, one ankle brace, one foot brace, one tendon-ligament-skin architecture, and four PMAs. The tendon-ligament-skin and four PMAs correspond to muscles, mimicking the anatomy of the human's ankle and foot. The four PMAs mimic both the morphology and the functionality of biological muscles to control dorsiflexion, plantar flexion, inversion, and eversion of the ankle (Nagpal, 2014). However, the bionic structure of this CDAD is complicated, which limits its wider applications.

2.3 Requirements of Robotic Assistive Devices

Several papers that review the research and development of RADs have been published recently, covering the assistance of the full spectrum of human extremities, including upper limbs (Babaiasl et al., 2016; Basteris et al., 2014; Gopura et al., 2016; Jarrassé et al., 2014; Lo & Xie, 2012; Loureiro, Harwin, Nagai, & Johnson, 2011; Maciejasz et al., 2014; Masiero & Armani, 2011; Morales et al., 2011; Niyetkaliyev et al., 2017; Norouzi-Gheidari, Archambault, & Fung, 2012; Proietti et al., 2016), lower limbs (Belda-Lois et al., 2011; G. Chen, Chan, Guo, & Yu, 2013; Huo et al., 2016; Hussain, 2014; Jamwal, Hussain, & Xie, 2013; Pennycott, Wyss, Vallery, Klamroth-Marganska, & Riener, 2012; Sánchez, Díaz, & Gil, 2011; Yan et al., 2015), and hands (Balasubramanian, Klein, & Burdet, 2010; Heo, Gu, Lee, Rhee, & Kim, 2012; Lum, Godfrey, Brokaw, Holley, & Nichols, 2012). Requirements for general RADs are applicable to CDADs. Popular requirements on safety, weight, adaptability, versatility, and misalignment are discussed in chapter two.

2.3.1 Safety

Since RADs are usually in direct contact with people, their safety in use is paramount (Tsagarakis & Caldwell, 2003). Researchers have been working mainly from the following three aspects in order to improve the safety of RADs.

- Design optimization. When a RAD is at a singular pose, it is out of control, which could lead to injuries to people. Thus, singularities of the RAD must be avoided within the workspace of the RAD (Gopura et al., 2016). Singularities within the workspace of the RAD can be eliminated or avoided by optimizing the design of the RAD.
- Inherent safety. Inherent safety means the design of a RAD can guarantee the safety of the people to some extent, even in case of failure (Veneman, Ekkelenkamp, Kruidhof, van der Helm, & van der Kooij, 2006). Actuators with inherent safety, such as PMAs (Tsagarakis & Caldwell, 2003) and SEAs (Veneman et al., 2006), have been applied on RADs to achieve the inherent safety.
- Back-drivability. Back-drivability of a RAD allows a person with limited mobility to override the motion of the RAD and back-drive the RAD (J. Li et al., 2012; Mao & Agrawal, 2012). Back-drivability can improve the safety of the people (Maciejasz et al., 2014).

2.3.2 Weight

The weight requirement of a RAD depends on its specific application. If the RAD is used in a fixed location (e.g., a hospital), the mass of the non-movable parts (e.g., the base fixed on the floor) of the RAD is usually not a problem, but the mass of the movable parts moving with a person with limited mobility could be an issue because movable parts require the use of actuators to counterbalance gravity effects (Tsagarakis & Caldwell, 2003) and control their motions. However, if the RAD needs to be portable, the total weight of the RAD is critical. The portability of a RAD requires the RAD to be compact, lightweight, and energy-efficient (Moreno, Brunetti, Navarro, Forner-Cordero, & Pons, 2009; Pons, 2010). A portable RAD makes it convenient for a person with limited mobility to support his or her ADLs. Ambulatory platforms (e.g., a wheelchair) can be used to support RADs and make them ambulatory (Pedrocchi et al., 2013). Nonetheless, no matter the RAD is carried by the person with limited mobility or an ambulatory platform, lightweight allows the RAD to be applied in more scenarios.

2.3.3 Adaptability

People vary in size, shape, and motor ability, which presents a tremendous challenge to the design of RADs (S. Ball, Brown, & Scott, 2007). Robotic assistive devices have to be adaptive to accommodate people with various demographic and anthropomorphic parameters (Riener, 2007).

For RADs with rigid linkages, the adaptability is challenging. The lengths of their linkages need to be adjustable to fit various people to make the RADs adaptive to various people (S. J. Ball, Brown, & Scott, 2007). However, using linkages with adjustable length inevitably increases the mass of the RAD and the setup time of each wearing.

2.3.4 Versatility

Human joints (e.g., human wrist and human shoulder) have multiple DOFs, which makes human joints versatile and flexible, but presents design challenges for RADs. A robotic joint of a RAD can employ multiple single-DOF revolute axes to mimic a multi-DOF human joint. Considering space limitation, it is challenging to have a compact design of such a robotic joint.

2.3.5 Misalignment

Joint misalignment between a person and a RAD occurs during the setup of a RAD (Esmaeili, Gamage, Tan, & Campolo, 2011). A robotic joint (e.g., robotic shoulder) and a human joint (e.g., human shoulder) are misaligned if any revolute axis of the robotic joint doesn't align perfectly with the corresponding anatomical axis of the human joint (Loureiro et al., 2011). Joint misalignment can cause excessively joint forces and torques, resulting in discomforts or even injuries (e.g., excessive joint wear) to people (Gopura et al., 2016). Various techniques have been used in RADs to minimize joint misalignment. For example, MEDARM employed additional revolute axes to allow the RAD to shift its revolute axes to mitigate the misalignment problem (S. J. Ball et al., 2007). Moreover, NEUROExos (Vitiello et al., 2013) utilized a special mechanism that allows small relative adjustment of the revolute axis of a robotic elbow. However, additional mechanisms increase the weight and the complexity of RADs and thus, are not practical for portable RADs (Gopura et al., 2016).

2.4 Pros and Cons of Cable-Driven Assistive Devices

A robotic joint of a CDAD is formed by a CDPR. The robotic joint has no physical revolute axes and thus, the pros and cons of a CDAD are special.

2.4.1 Benefits of Cable-Driven Assistive Devices

A CDAD has several benefits. Firstly, a CDAD has adaptability because it does not have any rigid linkage and a robotic joint is free of revolute axes. Thus, a CDAD can be worn by people with various sizes. If the controller of the CDAD can also adapt to the various sizes of people, additional adjustment mechanisms for various people are not necessary anymore. In this case, the time spent on setting up the CDAD can be reduced as well.

Secondly, a robotic joint of the CDAD can provide multiple DOFs for a human joint via a proper number of cables. Thus, the CDAD has great versatility and does not restrict the natural DOFs of a human joint (Gopura et al., 2016). Moreover, formed by a CDPR, a robotic joint of a CDAD does not need to employ multiple revolute axes to provide multiple DOFs for a human joint, such as human shoulder and human wrist.

Thirdly, a CDAD does not have rigid linkages, which makes it lighter. The shape and the mass of a CDAD do not increase significantly if the number of DOFs of its robotic joint is increased. Only additional cables, rather than linkages, are needed to provide additional DOFs. If soft braces are utilized to replace cuffs, the mass of a CDAD can be further reduced. Making a CDAD light and compact enough to be worn by a person to support ADLs is possible.

Fourthly, a CDAD does not have the misalignment issue between the revolute axes of the device and the anatomical axes of the people. The reason is that the robotic joint of a CDAD is formed by a CDPR which does not have any revolute axis.

Fifthly, if SEAs, PMAs, or soft braces are utilized on a CDAD, SEAs, PMAs, or soft braces can provide the inherent safety. In general, a CDAD can meet all five rehabilitation requirements discussed above.

2.4.2 Drawbacks of Cable-Driven Assistive Devices

A CDAD also has drawbacks. Firstly, it is challenging to control a CDAD. The robotic joint of a CDAD is formed by a CDPR. Modeling and control methods of a CDPR usually assume the end-effector of the CDPR has no collision with cables. However, such a cable-collision-free

assumption is not always true for a CDAD. Moreover, when soft braces are utilized, anchor points (Merlet, 2010) of a CDAD are possible to be close to or even in contact with segments of a person. The collision between cables and segments of a person is inevitable in this case. Thus, traditional modeling and control methods of a CDPR cannot be applied to CDADs if collisions occur between segments of a person and cables. Moreover, uncertainties (e.g., uncertain positions of anchor points due to the wearing inconsistency for each training) make the control of the CDAD even more challenging.

Secondly, the most distinguished characteristic of a CDPR is the unidirectional constraint of cables (i.e., cables can pull but cannot push) (Xiong & Diao, 2017a). Hence, a robotic joint of a CDAD formed by a CDPR cannot be pressed. The robotic joint applies additional pressure to a human joint when assisting a human joint (Xiong, Zhang, Liu, & Diao, 2018). Such a drawback of the robotic joint formed by a CDPR might constrain the application of the CDAD.

2.5 Artificial Neural Networks

Artificial neural networks are models for solving intelligent tasks by machines inspired by biological neural networks (Yegnanarayana, 2009). The ANN is considered as a "black-box" as it imitates a behavior rather than a structure from the system engineering point of view (Prieto et al., 2016). A benefit of the "black box" is that, it is able to describe virtually any non-linear dynamics (Prieto et al., 2016). ANNs have proven to be competitive in the resolution of non-linear problems (L. Jin, Zhang, & Li, 2016; D. Lin, Wang, Nian, & Zhang, 2010; Xiao & Liao, 2016). The modification of the parameters of the ANN carries out an adaptation to the environment via learning (Prieto et al., 2016). Using ANNs for controlling robots have attracted much attention in recent years (L. Jin, Li, Yu, & He, 2018). Several types of ANNs, such as Recurrent Neural Networks (RNNs), Feedforward Neural Networks (FNNs), and Deep Neural Networks (DNNs), have been used in the control of robotics.

2.5.1 Feedforward Neural Networks

Feedforward Neural Networks with multiple layers is composed of one or more hidden neural layers (Da Silva, Spatti, Flauzino, Liboni, & dos Reis Alves, 2017). The diagram of an example FNN with one hidden layer is shown in Figure 2.3. The FNN is the most popular type of ANNs

due to the flexibility in structure, representational capabilities, and several available training algorithms (Sharma & Chandra, 2010). FNNs are capable of approximating nonlinear functions to any desired degree of accuracy (Anastassiou, 2011; Z. Chen, Cao, & Hu, 2015).

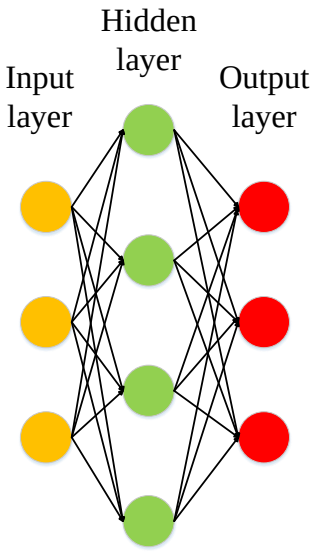


Figure 2.3 Diagram of a FNN

2.5.2 Recurrent Neural Networks

Recurrent Neural Networks can be applied to time-variant systems, such as system identification, time series prediction, process control, and so forth (Da Silva et al., 2017). The diagram of an example RNN is shown in Figure 2.4.

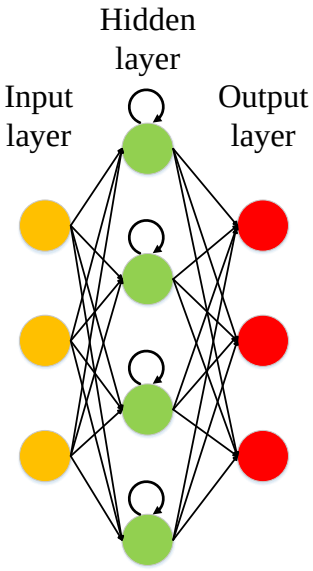


Figure 2.4 Diagram of a RNN

2.5.3 Deep Neural Networks

The concept of deep learning algorithms was proposed in the late 20th century inspired by deep hierarchical structures of human speech perception and production systems (Liu et al., 2017). Hinton (Hinton, Osindero, & Teh, 2006) developed a novel training method giving the birth of deep learning techniques in 2006. Either FNNs or RNNs can form DNNs with deep hierarchical structures, as shown in Figures 2.5 and 2.6. The DNNs are built from a cascade of hidden layers of units between the input and output layers (Schmidhuber, 2015). If the numbers of layers and units are increased, the DNNs can represent functions with higher complexity (Liu et al., 2017).

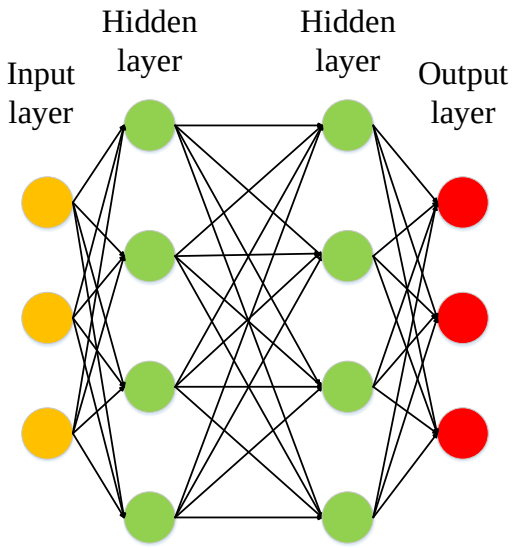


Figure 2.5 Diagram of a deep hierarchical FNN

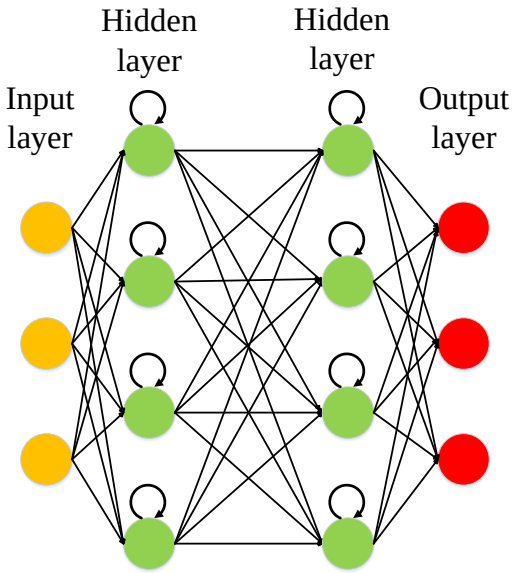


Figure 2.6 Diagram of a deep hierarchical RNN

2.5.4 Applications of Artificial Neural Networks on Cable-Driven Robots

Artificial neural networks have been applied to the studies of cable-driven robots. The forward kinematics problem of a CDPR was solved by a FNN (Ghasemi, Eghtesad, & Farid, 2010). Moreover, the inverse kinematics problem of a soft robot controlled by three cables was discussed in a reference (Giorelli, Renda, Ferri, & Laschi, 2013). Authors of a reference (Giorelli et al., 2015)

compared a Jacobian-based method with a FNN in solving the inverse kinematics problem of a cable-driven soft robot with two cables in terms of accuracy and computational time based on experiments.

2.6 Deep Reinforcement Learning

Deep reinforcement learning can extract features from high-dimensional data and learn complex policies by introducing DNNs to solve reinforcement learning problems. The DRL is suitable for various tasks (Tsurumine, Cui, Uchibe, & Matsubara, 2019), especially for sequential decision-making applications. Several DRL algorithms, such as DQN (Mnih et al., 2013), Trust Region Policy Optimization (TRPO) (Schulman, Levine, Abbeel, Jordan, & Moritz, 2015), Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015), and Mirror Descent Guided Policy Search (MDGPS) (Montgomery & Levine, 2016), have recently been used to solve various manipulation tasks in robotics (F. Li, Jiang, Zhang, Wei, & Song, 2019; Z. Li, Xue, Lin, & Tong, 2018; A Nagabandi, Kahn, Fearing, & Levine, 2018; Passalis & Tefas, 2019; Rahman, Rashid, & Hossain, 2018; M. Zhang et al., 2017; T. Zhang, Kahn, Levine, & Abbeel, 2016).

2.6.1 Deep Deterministic Policy Gradient

Since the DDPG was proposed by (Lillicrap et al., 2015) to solve reinforcement learning problems with continuous state and action, it is appropriated for a CDAD to learn to assist a human joint. The DDPG uses an actor-critic architecture combined with the DQN (Mnih et al., 2016) and the Deterministic Policy Gradient (DPG) (Silver et al., 2014). The DDPG utilizes four DNNs (i.e., actor network, actor-target network, critic network, and critic-target network) to approximate two policies (i.e., behavior policy and target policy), as shown in Table 2.1.

Table 2.1 Networks of DDPG

Network	**Input**	**Output**	**Policy**
Actor Network	$\boldsymbol{s}_t$	$\boldsymbol{a}_t = \boldsymbol{\mu}(\boldsymbol{s}_t \mid \theta^{\mu})$	Behavior Policy
Actor-Target Network	$\boldsymbol{s}_{t+1}$	$\boldsymbol{a}'_t = \boldsymbol{\mu}'(\boldsymbol{s}_{t+1} \mid \theta^{\mu'})$	Target Policy
Critic Network	$\boldsymbol{s}_t, \widetilde{\boldsymbol{a}}_t$	$Q = Q(\boldsymbol{s}_t, \widetilde{\boldsymbol{a}}_t \mid \theta^{Q})$	Behavior Policy
Critic-Target Network	$\boldsymbol{s}_{t+1}, \boldsymbol{a}'_t$	$Q' = Q'(\boldsymbol{s}_{t+1}, \boldsymbol{a}'_t \mid \theta^{Q'})$	Target Policy

The actor network of DDPG approximates a behavior policy $\boldsymbol{\mu}$, and the output of the actor network is (Lillicrap et al., 2015)

$$\boldsymbol{a}_t = \boldsymbol{\mu}(\boldsymbol{s}_t|\, \theta^{\mu}) \tag{2.12}$$

where t represents a specific time step. $\boldsymbol{s}_t$ represents the state in time step t. θ^{μ} is the actor network parameter. The policy function uses deterministic policy instead of stochastic policy. The critic network of DDPG approximates a value function as (Lillicrap et al., 2015)

$$Q = Q(\boldsymbol{s}_t, \widetilde{\boldsymbol{a}}_t|\, \theta^{Q}) \tag{2.13}$$

θ^{Q} is the critic network parameter. $\widetilde{\boldsymbol{a}}_t$ can be expressed as

$$\widetilde{\boldsymbol{a}}_t = \boldsymbol{a}_t + \boldsymbol{N}_t \tag{2.14}$$

where $\boldsymbol{N}_t$ is an exploration noise.

The actor-target network approximates a target policy $\boldsymbol{\mu}'$ and the output of the actor-target network is (Lillicrap et al., 2015)

$$\boldsymbol{a}'_t = \boldsymbol{\mu}'(\boldsymbol{s}_{t+1}|\, \theta^{\mu'}) \tag{2.15}$$

where $\theta^{\mu'}$ is the actor-target network parameter. The critic-target network approximates a value function (Lillicrap et al., 2015)

$$Q' = Q'(\boldsymbol{s}_{t+1}, \boldsymbol{a}'_t|\, \theta^{Q'}) \tag{2.16}$$

where $\theta^{Q'}$ is the critic-target network parameter.

The actor network is updated with respect to the gradient of expected performance objective J as

$$\nabla_{\theta^{\mu}} J \approx \mathbb{E}_{\boldsymbol{s}_t \sim \rho^{\beta}}[\nabla_a Q(\boldsymbol{s}, \boldsymbol{a}|\theta^{Q})|_{\boldsymbol{s}_t, \boldsymbol{\mu}(\boldsymbol{s}_t)} \nabla_{\theta^{\mu}} \boldsymbol{\mu}(\boldsymbol{s}|\, \theta^{\mu})|_{\boldsymbol{s}_t}] \tag{2.17}$$

where performance objective J is defined as

$$J = \mathbb{E}_{\boldsymbol{s}_t \sim \rho^{\beta}}[Q(\boldsymbol{s}, \boldsymbol{a}|\theta^{Q})|_{\boldsymbol{s}_t, \boldsymbol{\mu}(\boldsymbol{s}_t)}] \tag{2.18}$$

where ρ^{β} is a state distribution. The critic network is updated by minimizing the loss L, defined as

$$L(\theta^{Q}) = \mathbb{E}_{\boldsymbol{s}_t \sim \rho^{\beta}, \boldsymbol{a}_t \sim \beta, r \sim E}[(y_t - Q(\boldsymbol{s}, \boldsymbol{a}|\theta^{Q})|_{\boldsymbol{s}_t, \boldsymbol{\mu}(\boldsymbol{s}_t|\, \theta^{\mu})})^2 \tag{2.19}$$

where E is the environment and β is an action distribution. y_t can be expressed as

$$y_t = r_t + \gamma Q'(\boldsymbol{s}_{t+1}, \boldsymbol{\mu}'(\boldsymbol{s}_{t+1}|\theta^{\mu'})|\theta^{Q'}) \tag{2.20}$$

where $\gamma \in (0,1)$ denotes a discount factor. r_t is a reward function. After updating actor and critic networks, two target networks are updated by

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \tag{2.21}$$

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \tag{2.22}$$

where update rate $\tau \ll 1$. The DDPG also utilizes experience replay buffer to avoid correlations in sequences. A mini-batch is randomly selected from the replay buffer for the training of DDPG in each time step.

2.7 Summary

A comprehensive review of CDPRs, CDADs, RADs, ANNs, and DRL is presented in chapter two. A CDAD meets the requirements of RADs, such as safety, weight, adaptability, versatility, and misalignment. But the control of a CDAD is challenging. Thus, ANNs and DRL, which have been used to solve a variety of manipulation tasks in robotics, are taken into account in this study to control a CDAD.

CHAPTER 3. RESEARCH METHODOLOGY

According to the literature review in chapter two, a CDAD meets the requirements of RADs such as safety, weight, adaptability, versatility, and misalignment. However, the control of the CDAD is challenging because of the inconsistency of setups, the modularity of the CDAD, and the reconfiguration of the CDAD. Thus, learning control strategies are developed in this study for a CDAD that can adaptively assist a human joint. The author's work about fundamental studies of CDADs is demonstrated at the beginning of chapter three. Then, a PID–FNN control strategy and DDPG-based strategies are proposed to enable a CDAD to learn physical human-robot interactions and to control the pose of the human joint. Simulations are designed to validate the learning control strategies. Sections 3.1.1, 3.1.2, 3.2.1, and 3.2.2 of chapter three are revised based on the author's published papers (Xiong & Diao, 2018a), (Xiong et al., 2018), (Xiong, Zhang, & Diao, 2019), and (Xiong, Ma, Zhang, & Diao, 2019), respectively.

3.1 Fundamental Studies of Cable-Driven Assistive Devices

3.1.1 Workspace Isotropy

Workspace isotropy analysis in terms of the geometric shape of the workspace is crucial for many applications in which the base of a parallel manipulator (e.g., CDPR) is movable. To better understand and analyze workspace isotropy for parallel manipulators, a clear definition of workspace isotropy is essential. A robotic manipulator is said to be isotropic if its end-effector has the same properties in any arbitrary direction (Gosselin & Angeles, 1989). The properties include, but are not limited to, characteristics of kinematics, dynamics, force, and workspace. The study proposes workspace isotropy indices to evaluate the workspace of a parallel manipulator in terms of its geometric shape. The workspace of a parallel manipulator is geometrically isotropic about a selected point, called isotropy analysis center (IAC), if, with an arbitrary rotation applied to the movable base of the parallel manipulator about the IAC, the workspace after the rotation overlaps the workspace before the rotation. The study assumes that the base of a parallel manipulator is movable and that the workspace of the parallel manipulator rotates, but does not deform, with the rotation of the movable base. The workspace of a parallel manipulator should be continuous, rather than a union of a few separated spaces.

The concept of isotropy of a robotic manipulator has been discussed by many researchers. Gosselin et al. studied the symmetry of planar parallel manipulators (Gosselin & Angeles, 1989). The mobility isotropy of a mobile robot was investigated in (S. Kim, Kim, & Moon, 2005). Rosati et al. studied how to evenly distribute the inertia of planar robotic manipulators designed for upper-limb rehabilitation (Rosati, Secoli, Zanotto, Rossi, & Boschetti, 2008) while others proposed isotropy indices, such as tension factor (Pham, Yeo, Yang, & Chen, 2009) and force index (Rosati, Zanotto, & Agrawal, 2011) concerning isotropy of cable tensions in CDPRs. Salcudean and Stocco studied the force isotropy of haptic devices to achieve a uniform force capability matching that of the human hand (Salcudean & Stocco, 2000). The kinematics isotropy of serial manipulators (Gosselin & Lavoie, 1993) and rigid-link parallel manipulators (Fattah & Ghasemi, n.d.; Zanganeh & Angeles, 1997) was addressed by researchers based on the condition number of the Jacobian matrix (Salisbury & Craig, 1982). Moreover, Hwang et al. discussed the kinematics isotropy of a haptic device (Yoon, Ryu, & Hwang, 2010) based on a global conditioning index (Gosselin & Angeles, 1991). Although the geometric shapes of the workspaces of various haptic device designs were compared in (Yoon et al., 2010) via Euler angles, no quantitative isotropy index was defined to evaluate workspace isotropy in terms of geometric shape. To the author's best knowledge, this is the first time to quantitatively study the workspace isotropy of parallel manipulators in terms of the geometric shape of the workspace.

3.1.1.1 Workspace Isotropy Indices

The IAC is a reference point selected for workspace isotropy analysis. It is not necessary the geometric center of the workspace. It even does not need to be inside the workspace. For a CDPR, the IAC could be set at the centroid of the polyhedron formed by the anchor points on the base if no specific position is of interest in workspace isotropy analysis. For a rigid-link parallel manipulator, the IAC could be set at the centroid of the end-effector when all joints are in the middle of their motion ranges. Alternatively, if one is interested in how isotropic a workspace is around a specific point, then the IAC should be set at that point. Different IACs generally lead to different values of the workspace isotropy indices.

The distance of a pose of a parallel manipulator is defined as the distance between the position of the pose and the IAC. For the study, a parallel manipulator is studied with three reference frames, namely, base frame, end-effector frame, and inertial frame. The base frame is mounted on the base of the parallel manipulator. The origin of the base frame is always set at the IAC. The end-effector frame is mounted on the end-effector of the parallel manipulator. The inertial frame is mounted on the ground. Both the base frame and the end-effector frame are movable while the inertial frame is fixed. The workspace of the parallel manipulator is described in the base frame and the motion of the base in the inertial frame. The inertial frame coincides with the base frame when there is no rotational disturbance to the movable base of the parallel manipulator. Given a rotational disturbance to the movable base of the parallel manipulator, the base frame and the workspace rotate while the inertial frame does not. For a general parallel manipulator, assuming the position of a pose in the base frame is described by three Cartesian coordinates *x*, *y*, and *z*, then the distance of this pose, denoted by r, is mathematically defined as

$$r = \sqrt{x^2 + y^2 + z^2} \tag{3.1}$$

For a planar parallel manipulator, $z = 0$. One can see from (3.1) that the distance of a pose depends on the position of the pose only.

The translational workspace of a parallel manipulator is defined as a set of positions that the parallel manipulator can physically reach at an arbitrary orientation with or without additional constraints (e.g., force-closure, singularity-free, etc.). The translational workspace is isotropic if, after an arbitrary rotation of the movable base of the parallel manipulator about the IAC, the new translational workspace completely overlaps with the original one. To quantitatively measure how isotropic the translational workspace of a parallel manipulator is, TWII is defined as

$$\text{TWII} = \frac{\int_{r_{min}}^{r_{max}} \eta_t dr}{r_{max} - r_{min}} \tag{3.2}$$

where r is the distance of a pose in the translational workspace. r_{min} and r_{max} are the minimum and the maximum distances of poses in the translational workspace, respectively. If the IAC is inside the translational workspace, then $r_{min} = 0$. η_t represents the isotropy of the translational workspace intersecting with a surface denoted as s_r. All positions on surface s_r have the same distance r ($r_{min} \leq r \leq r_{max}$) to the IAC. In other words, surface s_r is a spherical surface for a

three-dimensional translational workspace or a circle for a two-dimensional translational workspace with the IAC at its geometric center. η_t is defined as

$$\eta_t = \frac{\int_{s_r} t_d d_{s_r}}{A_{s_r}} \tag{3.3}$$

where A_{s_r} is the area of surface s_r for a three-dimensional translational workspace or the circumference of the corresponding circle for a two-dimensional translational workspace. ds_r is the infinitesimal of s_r. t_d is a factor to tell whether a ds_r is in the translational workspace.

$$t_d = \begin{cases} 1, \text{if } ds_r \text{ is in the translational workspace} \\ 0, \text{if } ds_r \text{ is not in the translational workspace} \end{cases} \tag{3.4}$$

Therefore, the numerator in (3.3) is the surface area of surface s_r or the arc length of the corresponding circle intersecting with the translational workspace. Thus, one has

$$0 < \eta_t \leq 1 \tag{3.5}$$

If all the poses with a distance of r are in the translational workspace, namely, surface s_r is entirely embedded in the translational workspace, then $\eta_t = 1$. According to the definition of TWII in (3.2), η_t with different distances have the same weight. Therefore, TWII is the arithmetic mean of η_t with the distance ranging from r_{min} to r_{max}.

Based on the definition, TWII ranges from 0 to 1. The larger the TWII, the more isotropic the translational workspace of the parallel manipulator. The geometric interpretation of TWII is the arithmetic mean of ratios of the total cross-section area to the total area of all spheres intersecting with the translational workspace if the translational workspace is three-dimensional. If the translational workspace is two-dimensional, TWII is the arithmetic mean of ratios of the total cross-section arc length to the total circumference of all circles intersecting with the translational workspace. Thus, if the translational workspace is spherical for a spatial parallel manipulator or circular for a planar one, then the translational workspace is isotropic (i.e., TWII = 1) if the IAC is selected at the geometric center of the translational workspace.

By inspecting how isotropic a translational workspace is, TWII can be used to evaluate the robustness of the translational workspace of a parallel manipulator to rotational disturbances about the IAC to the movable base. In other words, TWII is an indicator to a parallel manipulator's capability of returning to its original position when a rotational disturbance about the IAC is applied to the movable base of the parallel manipulator. It should be noted that the translational

workspace is isotropic (i.e., TWII = 1) is a necessary, but not sufficient, condition for a parallel manipulator to be able to compensate rotational disturbances and return to its original position.

The rotational workspace of a parallel manipulator is defined as a set of orientations that the parallel manipulator can physically reach at an arbitrary position with or without additional constraints (e.g., force-closure, singularity-free, etc.). The rotational workspace is isotropic if, after an arbitrary rotation of the base of the parallel manipulator about the IAC, the new rotational workspace completely overlaps with the original one. The more isotropic the rotational workspace, the better chance the manipulator returns to its original orientation after a rotation of its base. It should be pointed that the isotropy of the rotational workspace has nothing to do with the positions or the distances of poses, because the rotational workspace of a manipulator depends on its reachable orientations only. To quantitatively measure how isotropic the rotational workspace of a parallel manipulator is, RWII is defined as

$$\text{RWII} = \frac{w_r}{p_r} \tag{3.6}$$

where w_r is the volume, the area, or the orientation range of the rotational workspace when the rotational workspace is three-, two-, or one-dimensional, respectively. For example, if the rotational workspace is one-dimensional and the orientation of the parallel manipulator changes from -30 degrees to 50 degrees, then w_r =80 degrees in this case. p_r is the volume, the area, or the orientation range of the RWII-analysis space which includes all possible orientations of the manipulator. For example, if the RWII-analysis space is one-dimensional, then p_r is 360 degrees. If the rotational workspace includes all possible orientations, then $w_r = p_r$ or $\text{RWII} = 1$.

Based on the definition, RWII ranges from 0 to 1. The larger the RWII, the more isotropic the rotational workspace of the parallel manipulator. The geometric interpretation of RWII is the volume, area, or length ratio of the rotational workspace to the RWII-analysis space. By inspecting how isotropic a rotational workspace is, RWII can be used to evaluate the robustness of the rotational workspace of a parallel manipulator to rotational disturbances to the movable base of the parallel manipulator. The rotational disturbances are not necessary about the IAC. In other words, RWII is an indicator to a parallel manipulator's capability of returning to its original orientation when a rotational disturbance occurs to the movable base of the parallel manipulator. The rotational workspace is isotropic (i.e., RWII = 1) is a necessary, but not sufficient, condition

for a parallel manipulator to be able to compensate rotational disturbances and return to its original orientation.

The entire workspace of a parallel manipulator is defined as a set of poses that the parallel manipulator can physically reach with or without additional constraints (e.g., force-closure, singularity-free, etc.). The entire workspace is isotropic if, after an arbitrary rotation of the movable base of the parallel manipulator about the IAC, the new entire workspace completely overlaps with the original one. For example, assume the entire workspace of the planar CDPR shown in Fig. 2a is isotropic with the IAC located at point *O*. The *xy* frame is the base frame and the $x_I y_I$ frame is the inertial frame. The solid red square and the dashed black square represent the end-effector and the base of the CDPR, respectively. The solid black lines are cables. Initially, the CDPR is at the pose (0,0,0) (unit: mm, mm, deg) in the inertial frame, as shown in Figure 3.1a. After a rotational disturbance about the IAC is applied to the base of the CDPR, the CDPR moves to a new pose, as shown in Figure 3.1b. The end-effector of the CDPR is now off the pose (0,0,0) in the inertial frame. Since the entire workspace of the CDPR is isotropic, it is possible for the CDPR to return to the original pose (0,0,0) in the inertial frame. As shown in Figure 3.1c on page 31, although the base of the CDPR is rotated due to a disturbance, the end-effector can still return to its original pose (0,0,0) in the inertial frame.

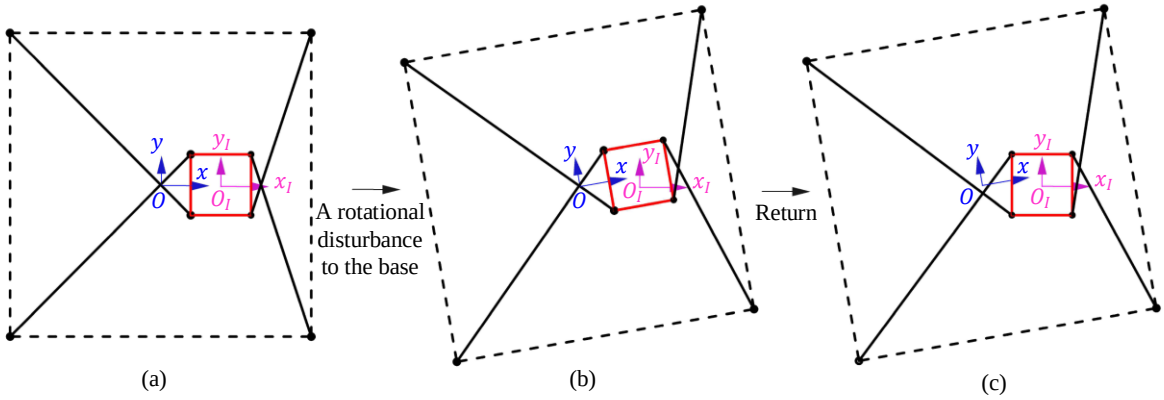


Figure 3.1 A planar CDPR with isotropic workspace returns to its original pose

The workspace of a parallel manipulator is isotropy if the parallel manipulator can reach the same poses in any arbitrary direction (Gosselin & Angeles, 1989). It should be noted that orientation references in different directions are different when discussing “the same poses” in the study of

workspace isotropy. Thus, the orientation references of different poses, whose positions are in different directions, are different in the study of workspace isotropy. In order to study "the same poses" with different orientation references and make "the same poses" have the same orientation with respect to different orientation references, local rotational workspace and adjusted local rotational workspace, denoted by a_l, are proposed. Specifically, the local rotational workspace at a position is the set of orientations that the parallel manipulator can physically reach at that position. Local rotational workspaces of all positions have the same orientation reference. For the study, local rotational workspaces of all positions share the same orientation reference with the entire workspace.

The adjusted local rotational workspace at a position is also the set of orientations that the parallel manipulator can physically reach at that position. However, the orientation reference of an adjusted local rotational workspace could be different from the local rotational workspace at the same position. The orientation reference of an adjusted local rotational workspace depends on the direction of the particular position. Thus, the orientation references of adjusted local rotational workspaces at different positions in different directions are different. With orientation references mentioned above, "the same poses" can have the same orientation with respect to the orientation references of their own adjusted local rotational workspace.

One example of illustrating the issue caused by using different orientation references is shown in Figure 3.2, where vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{a}'$ represent three different poses in the entire workspace of a planar parallel manipulator. The start point of a vector represents the position of a pose and the direction of a vector represents the orientation of the pose. The *xy* frame is the base frame and its origin, point *O*, is also the IAC. The $x_I y_I$ frame is the inertial frame and its origin is point O_I. The *x* axis of the base frame is used as the orientation reference for the entire workspace, rotational workspace and all local rotational workspaces. With *x* axis of the base frame as the orientation reference, poses $\boldsymbol{a}$ and $\boldsymbol{b}$ have the same orientation while poses $\boldsymbol{a}$ and $\boldsymbol{a}'$ have different orientations. However, if the base is rotated by an angle of β about the IAC at point *O*, pose $\boldsymbol{a}'$, rather than $\boldsymbol{b}$, overlaps with pose $\boldsymbol{a}$ before the rotation. Namely, $\boldsymbol{a}$ and $\boldsymbol{a}'$ are "the same poses" in different directions. A rotational workspace at a position with *x* axis of the base frame as the orientation reference is a local rotational workspace of that position.

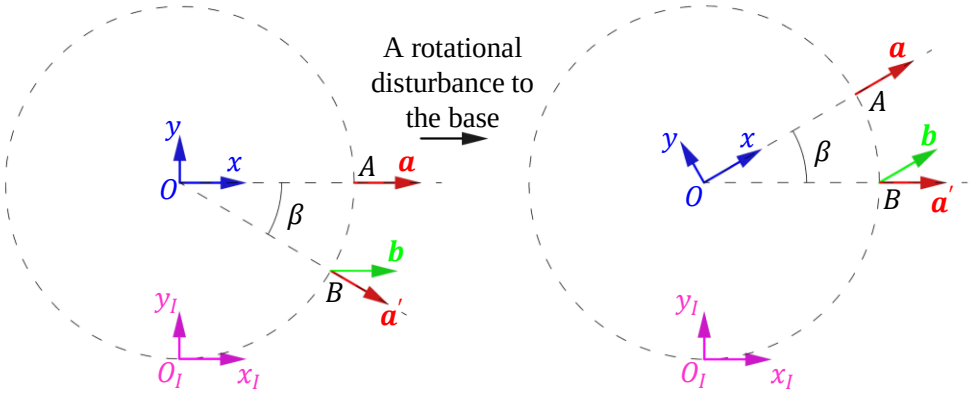


Figure 3.2 The effect of a position on its local rotational workspace for a planar CDPR

However, if the orientation reference of a pose is the vector from the IAC at point *O* to the position of the pose, namely, *OA* is the orientation reference for pose $\boldsymbol{a}$ and *OB* for poses $\boldsymbol{a}'$ and $\boldsymbol{b}$, poses $\boldsymbol{a}$ and $\boldsymbol{b}$ have different orientations while poses $\boldsymbol{a}$ and $\boldsymbol{a}'$ have the same orientation. In this case, "the same poses" (e.g., $\boldsymbol{a}$ and $\boldsymbol{a}'$) have the same orientation based on their own orientation references. A rotational workspace at a position with such an orientation reference is an adjusted local rotational workspace. EWII introduced below is based on such characteristics – "the same poses" in different directions have the same orientation with respect to orientation references defined by their positions. It should be pointed out that, since the method of defining the orientation reference by the vector from the IAC to the position of the pose for planar parallel manipulators is different from that for spatial ones, the adjusted local rotational workspace and EWII cannot be applied to spatial parallel manipulators.

To quantitatively measure how isotropic the entire workspace of a parallel manipulator is, EWII is defined as

$$\text{EWII} = \frac{\int_{r_{min}}^{r_{max}} \eta_w dr}{r_{max} - r_{min}} \tag{3.7}$$

$$\eta_w = \frac{\int_{s_r} \frac{V_{a_l}}{V_{a_r}} ds_r}{A_{s_r}} \tag{3.8}$$

where a_l is the adjusted local rotational workspace of a position on s_r. V_{a_l} is the volume of a_l. a_r is the space union set of the adjusted local rotational workspace a_l of all positions on s_r. V_{a_r} is the volume of a_r. Therefore, one has

$$0 \leq \frac{V_{a_l}}{V_{a_r}} \leq 1 \tag{3.9}$$

$$0 < \eta_w \leq 1 \tag{3.10}$$

According to the definition of EWII in (3.7), η_w with different distances have the same weight. Therefore, EWII is the arithmetic mean of η_w with the distance ranging from r_{min} to r_{max}.

Based on the definition, the EWII ranges from 0 to 1. The greater the EWII, the more isotropic the entire workspace of a parallel manipulator. The geometric interpretation of EWII is the arithmetic mean of distance-based entire workspace isotropy with distances ranging from r_{min} to r_{max}. EWII assesses the robustness of the entire workspace of a planar parallel manipulator to rotational disturbances about the IAC to the movable base of the manipulator. In another word, EWII can be used to evaluate a planar parallel manipulator's capability of returning to the original pose when a rotational disturbance about the IAC is applied to the movable base of the parallel manipulator. Again, it should be noted that the entire workspace is isotropic (i.e., EWII = 1) is a necessary, but not sufficient, condition for a parallel manipulator to be able to compensate rotational disturbances and return to its original pose.

3.1.2 Joint Force Analysis and Moment Efficiency

To gain the mobility of a human joint, a CDAD usually rotates the joint repeatedly by generating an assistant moment about the axis of the joint (Belda-Lois et al., 2011). In order to generate such an assistant moment during rehabilitation training, the CDAD needs to exert a force on the joint. Such a force may cause excessive wear of the joint or even break the joint if it exceeds a threshold. Thus, it is critical to analyze not only the assistant moment generated by the CDAD to rotate the joint, but also the joint force exerted by the CDAD on the joint for the sake of safety and the design of CDADs. Joint force issues have been discussed by a few researchers in the study of human-robot interaction for rehabilitation training. A method of trajectory optimization was proposed for a cable-driven upper arm exoskeleton to minimize the assistant moment and joint force in (Mao & Agrawal, 2010). A method of evaluating whether a joint force is safe was presented for human-care robots in (Ikuta, Ishii, & Nokata, 2003). The threshold of a safe joint force is defined in (Ikuta et al., 2003) as the minimal force that can cause injury to a human joint. However, a general analysis of the joint force generated by a CDAD is still not available in the literature. Such a joint

fore analysis can help evaluate the efficiency of the CDAD in providing assistant moment (called moment efficiency) with respect to generated joint force.

3.1.2.1 Moment Efficiency Index

Compared to the inertia of the end-effector, the inertia of cables is usually much smaller and can be ignored (Ottaviano & Castelli, 2010). Then, the motion of the end-effector can be expressed based on Newton-Euler Formulation as,

$$\boldsymbol{M}(\boldsymbol{X})\ddot{\boldsymbol{X}} + \boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)\dot{\boldsymbol{X}} + \boldsymbol{G}(\boldsymbol{X}) + \boldsymbol{F}_j(\boldsymbol{X}) = -\boldsymbol{J}^T\boldsymbol{\tau} \tag{3.11}$$

where $\boldsymbol{M}(\boldsymbol{X})$ denotes the mass matrix. $\boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)$ denotes the Coriolis and centripetal matrix. $\boldsymbol{G}(\boldsymbol{X})$ represents the 6×1 gravity vector. $\boldsymbol{F}_j(\boldsymbol{X})$ represents the 6×1 joint wrench vector. The mass matrix $\boldsymbol{M}(\boldsymbol{X})$, Coriolis and centripetal matrix $\boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)$, and the gravity matrix $\boldsymbol{G}(\boldsymbol{X})$ have been defined by several scholars (Khosravi & Taghirad, 2014b; Ma & Diao, 2005)

$$\boldsymbol{M}(\boldsymbol{X}) = \begin{bmatrix} m\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_p \end{bmatrix} \tag{3.12}$$

$$\boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)\dot{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{\omega} \times m\boldsymbol{v}_p \\ \boldsymbol{\omega} \times \boldsymbol{I}_p\boldsymbol{\omega} \end{bmatrix} \tag{3.13}$$

$$\boldsymbol{G}(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{F}_g \\ \boldsymbol{W}_g \end{bmatrix} \tag{3.14}$$

$$\boldsymbol{F}_j(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{F}_j \\ \boldsymbol{0}_{3\times1} \end{bmatrix} \tag{3.15}$$

where $\boldsymbol{v}_p$ is the translational velocity vector of the center of mass of the end-effector. $\boldsymbol{I}_p$ is the inertia tensor of the end-effector about point *O* in F_b. $\boldsymbol{I}_{3\times3}$ is the 3×3 identity matrix. $\boldsymbol{0}_{3\times3}$ is the 3×3 zero matrix. $\boldsymbol{0}_{3\times1}$ is the 3×1 zero vector. $\boldsymbol{F}_g$ is the gravity of the end-effector, and $\boldsymbol{W}_g$ is the moment of the gravity. $\boldsymbol{F}_j$ is the 3×1 joint force exerted on the joint by the end-effector. Moreover, one can write $\boldsymbol{J}^T\boldsymbol{\tau}$ as

$$\boldsymbol{J}^T\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{F}_t \\ \boldsymbol{W}_t \end{bmatrix} \tag{3.16}$$

where $\boldsymbol{F}_t$ is the resultant force of the cable tensions, and $\boldsymbol{W}_t$ is the resultant moment of the cable tensions.

Since the joint has three rotational DOFs, and the origin of the end-effector frame and the base frame are both set at the rotation center of the joint, one has $\dot{\boldsymbol{X}} = \begin{bmatrix}0\ 0\ 0\ \dot{\theta}_x\ \dot{\theta}_y\ \dot{\theta}_z\end{bmatrix}^T$ and $\ddot{\boldsymbol{X}} = \begin{bmatrix}0\ 0\ 0\ \ddot{\theta}_x\ \ddot{\theta}_y\ \ddot{\theta}_z\end{bmatrix}^T$. Then the last three equations of (3.11) can be rewritten as

$$\boldsymbol{I}_p\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}_p\boldsymbol{\omega} + \boldsymbol{W}_g = -\boldsymbol{W}_t \tag{3.17}$$

where $\dot{\boldsymbol{\omega}}$ is the derivative of $\boldsymbol{\omega}$. The first three equations of (3.11) can be rewritten as

$$\boldsymbol{F}_j = -\boldsymbol{F}_t - \boldsymbol{F}_g - \boldsymbol{\omega} \times m\boldsymbol{v}_p \tag{3.18}$$

where (3.17) determines the rotation of the end-effector, and the joint force can be calculated based on (3.18).

The joint force generated by the CDAD is $-\boldsymbol{F}_t$ and the joint force generated by the gravity of the end-effector is $-\boldsymbol{F}_g$. Moreover, the joint force generated by the centrifugal is $-\boldsymbol{\omega} \times m\boldsymbol{v}_p$. The study defines a moment efficiency index η as the magnitude of the assistant moment of the CDAD (i.e., $|\boldsymbol{W}_t|$) divided by the magnitude of the joint force generated by the CDAD (i.e., $|\boldsymbol{F}_t|$)

$$\eta = \frac{|\boldsymbol{W}_t|}{|\boldsymbol{F}_t|} \tag{3.19}$$

The characteristic length of η is set to 1 m in this study, so the unit of η is homogeneous (Nguyen & Gouttefarde, 2014). The moment efficiency index reflects the ability of a CDAD to assist a joint while leading to a unit magnitude of joint force acting on the joint. η varies according to the orientation of the joint and the design of the CDAD.

Antagonistic cable tensions (Xiong & Diao, 2017b, 2017a) of a CDAD are cable tensions that balance each other. Since a joint with three rotational DOFs can balance any translational force in theory, antagonistic cable tension vector of the CDAD with three rotational DOFs, noted as $\boldsymbol{\tau}_a$, satisfies

$$\boldsymbol{J}^T\boldsymbol{\tau}_a = \begin{bmatrix}\boldsymbol{F}_a \\ \boldsymbol{0}_{3\times1}\end{bmatrix} \tag{3.20}$$

Antagonistic cable tensions only balance the moment generated by each other. The resultant force generated by antagonistic cable tensions acts on the joint. It should be noted from (3.20) that, since antagonistic cable tensions create zero moment, they do not affect the rotation of the end-effector. However, antagonistic cable tensions do create a nonzero joint force, denoted as $\boldsymbol{F}_a$, acting on the joint. According to (3.20), in the force-closure workspace (Diao & Ma, 2007) of the CDAD,

increasing antagonistic cable tensions does not affect the wrench acting on the end-effector, but it does increase the magnitude of the joint force generated by the CDAD. Thus, based on (3.19), the moment efficiency index will decrease.

3.2 Learning Control Strategies Designed for Cable-Driven Assistive Devices

3.2.1 A Learning Control Framework Based on Artificial Neural Networks

The control of CDPRs has also been investigated by many researchers. PID controllers were widely used in the control of CDPRs (Alamdari & Krovi, 2015b; W. Chen et al., 2015; Cui et al., 2017; J. Lin & Liao, 2016; Xiong & Diao, 2018b; Zi, Ding, Cao, Zhu, & Kecskeméthy, 2014). The control of CDPRs based on optimal controllers (Jamshidifar et al., 2018; Korayem, Yousefzadeh, & Beyranvand, 2017) as well as fuzzy controllers (Wang et al., 2015) was also discussed by researchers. Alikhani and Vali proposed a sliding mode controller to control a large scale CDPR in (Alikhani & Vali, 2011). Adaptive controllers have a reliable performance in the control of CDPRs (Babaghasabha et al., 2015; Kino et al., 2018; Lamaury et al., 2013). An adaptive controller based on a ANN was used to control a CDPR in (Asl & Janabi-Sharifi, 2017). Singular perturbation approach was developed to control CDPRs with elastic cables in (Khosravi & Taghirad, 2014a). In (Caverly & Forbes, 2014), a passivity-based control method was applied to CDPRs. To deal with uncertainties of CDPRs, the robustness of controllers was studied (Babaghasabha et al., 2015; Jamshidifar et al., 2018; Khosravi & Taghirad, 2014b). The control of CDPRs with position-controlled actuators was discussed in (Begey, Cuvillon, Lesellier, Gouttefarde, & Gangloff, 2019). All the above-mentioned control strategies assumes the Jacobians (Diao & Ma, 2009) of the CDPRs can be determined in the control process. In other words, the above-mentioned control strategies are not applicable to the control of CDPRs with unknown Jacobians.

The Jacobian of a CDPR may be unknown in some applications. Take the CDPR in Figure 2.2 for an example. When setting up the CDPR for the rehabilitation of ankles, the positions and orientations of both the cuff and the brace may vary from person to person and from time to time, so do the positions of the anchor points. Since the Jacobian of a CDPR is determined by the positions of anchor points (Diao & Ma, 2008), the Jacobian of the CDPR in Figure 2.2 is not known

for each setup due to the inconsistency of each setup. Therefore, the control strategies based on known Jacobians cannot be directly applied to control the CDPR in Figure 2.2. Although one can measure the positions of the anchor points after each setup to determine the Jacobian, it is inconvenient.

To control CDPRs with unknown Jacobian, learning-based control strategies are taken into account. DRL is suitable for sequential decision-making applications (e.g., robot manipulation) without the knowledge of the system dynamics (Tsurumine et al., 2019). Many learning-based control strategies based on DRL algorithms, such as deep Q-network (DQN) (Mnih et al., 2013) and DDPG (Lillicrap et al., 2015) have recently been developed for a variety of robot manipulation problems. Physics simulators are wildly used to validate learning-based control strategies for robot manipulation problems. TRPO was used by (A Nagabandi et al., 2018) to control bio-inspired robots in the MuJoCo simulator (Todorov, Erez, & Tassa, 2012). Reference (Rahman et al., 2018) applied a DQN to control a self-balancing robot with two wheels in the Gazebo simulator (Koenig & Howard, 2004). Although DRL is powerful in solving robot manipulation problems and is promising in controlling CDPR with unknown Jacobian, DRL is lack of data efficiency and stability guarantee (Golemo, Taïga, Oudeyer, & Courville, 2018).

Artificial Neural Network is known for its universal approximation (Anastassiou, 2011; Huang, Yan, Zhou, & Xu, 2015; Y. Li & Wang, 2018; Yang & Yan, 2015) for a nonlinear system and has been applied to some CDPRs. An ANN has been used to approximate the forward kinematics of a CDPR in (Ghasemi et al., 2010). An ANN has been used in parallel with a controller to compensate for model uncertainties in controlling a CDPR in (Asl & Janabi-Sharifi, 2017).

3.2.1.1 A Learning Control Framework Based on Artificial Neural Networks

A novel control framework for a CDAD is proposed, concerning uncertainties of parameters in each setup of the CDAD. The control framework has a robust controller and an ANN. The robust controller controls the device in task space by calculating a wrench that is supposed to be applied to the device. The ANN is able to map the wrench applied on the end-effector at a pose in task space to a set of cable tensions in joint space based on training. Then, one can obtain a set of cable tensions corresponding to a wrench at a pose using this mapping, even the setup the device is

different from case to case. Therefore, one can control the CDAD. Such a mapping can be expressed as

$$\boldsymbol{\tau} = ANN(\boldsymbol{w}, \boldsymbol{X}) \tag{3.21}$$

where $\boldsymbol{\tau}$ is the vector of cable tensions that can generate a wrench $\boldsymbol{w}$ at the current pose $\boldsymbol{X}$. The inputs of the ANN are the wrench $\boldsymbol{w}$ required to control the CDAD in task space and the current pose $\boldsymbol{X}$ of the CDAD. The ANN outputs a cable tension vector $\boldsymbol{\tau}$ that contains a set of cable tensions that can generate the wrench $\boldsymbol{w}$ at the current pose $\boldsymbol{X}$.

The flow diagram of the novel control framework is shown in Figure 3.3. When a CDAD is controlled to move from the current pose to a target pose, the robust controller calculates the target wrench required to achieve such a motion based on the dynamics model of the CDAD. The ANN then maps the target wrench to be applied on the end-effector at current pose in task space to a set of cable tensions in joint space according to (3.21). The actuators finally deliver the target cable tensions and drive the CDAD to the target pose.

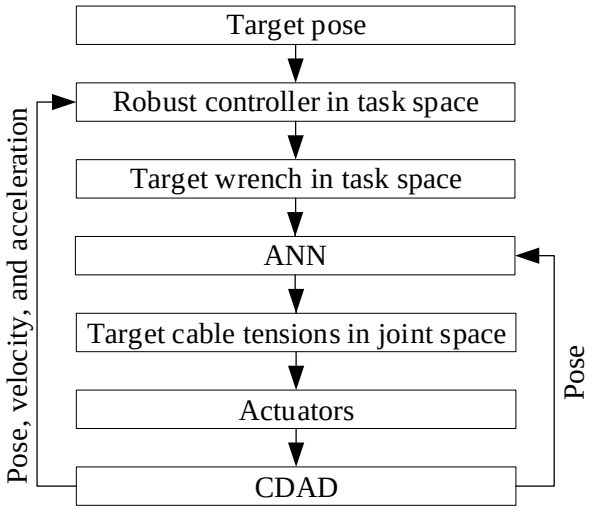


Figure 3.3 Diagram of a control framework for CDADs

Various ANNs, such as FNNs (Anastassiou, 2011; Z. Chen et al., 2015; Costarelli & Spigler, 2013), RNNs (Funahashi & Nakamura, 1993), or DNNs (Hinton & Salakhutdinov, 2006), can be used in this control frameworks. The robust controller can be a PID controller (Khosravi & Taghirad, 2014b), a sliding mode controller (Alikhani & Vali, 2011), a fuzzy controller (Wang et al., 2015),

or other robust controllers. Depending on the specific robust controller and ANN used in the control framework, various control strategies can be developed to control CDADs.

The ANN in the proposed control framework has to be trained before it is used to map a wrench applied on the end-effector at a pose in task space to a set of cable tensions in joint space. The ANN maps a wrench $\boldsymbol{w}$ at a pose $\boldsymbol{X}$ to a cable tension vector $\boldsymbol{\tau}$. Thus, data of wrenches, poses, and cable tensions are necessary to be collected to train the ANN.

The study assumes that the disturbance wrench caused by model uncertainties (e.g., mass uncertainty and moment of inertia uncertainty) is denoted by $\boldsymbol{w}_d$. Moreover, an approximation error is inevitable when the ANN maps the target wrench at the current pose to the set of target cable tensions. Namely, the set of target cable tensions derived via the ANN does not generate the target wrench exactly. Compared to the target wrench computed by the robust controller, the wrench generated by the set of target cable tensions derived via the ANN has an error, denoted as $\boldsymbol{w}_N$. Then, the dynamics equation of the CDAD can be rewritten as

$$\boldsymbol{M}(\boldsymbol{X})\ddot{\boldsymbol{X}} + \boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)\dot{\boldsymbol{X}} + \boldsymbol{G}(\boldsymbol{X}) + \boldsymbol{w}_e + \boldsymbol{w}_d + \boldsymbol{w}_N = \boldsymbol{w} \tag{3.22}$$

3.2.1.2 A Learning Control Strategy Based on a Feedforward Neural Network

To illustrate how to use the proposed control framework to control a fully-constrained CDPR with unknown Jacobian, a control strategy, named PID–FNN control strategy, is developed in this section. A PID controller and an FNN with one hidden layer are employed for the robust controller and the ANN in the proposed control framework, respectively. Different from the conventional control strategies for CDPRs with known Jacobians (e.g., (Khosravi & Taghirad, 2014b)), the PID–FNN control strategy uses the FNN to map a wrench applied on the end-effector at a pose in the task space to a set of cable tensions in the joint space, rather than solving the inverse dynamics equation.

The architecture of the FNN used in the PID-FNN control strategy for a CDPR with *p* DOFs and *n* cables is shown in Figure 3.4. The FNN is a fully connected ANN. Since the CDPR has *p* DOFs, a wrench applied on the CDPR has *p* elements and a pose of the CDPR has *p* elements as well. There are *n* cable tensions since the CDPR has *n* cables. Therefore, the FNN has 2*p* inputs and *n*

outputs. To map a wrench applied on the end-effector at a pose in the task space to a set of cable tensions in the joint space, one just needs to input the p wrench elements of $\boldsymbol{w}$ and the p pose elements $\boldsymbol{x}$ to the ANN and the ANN can output n cable tension elements of the cable tension vector $\boldsymbol{\tau}$.

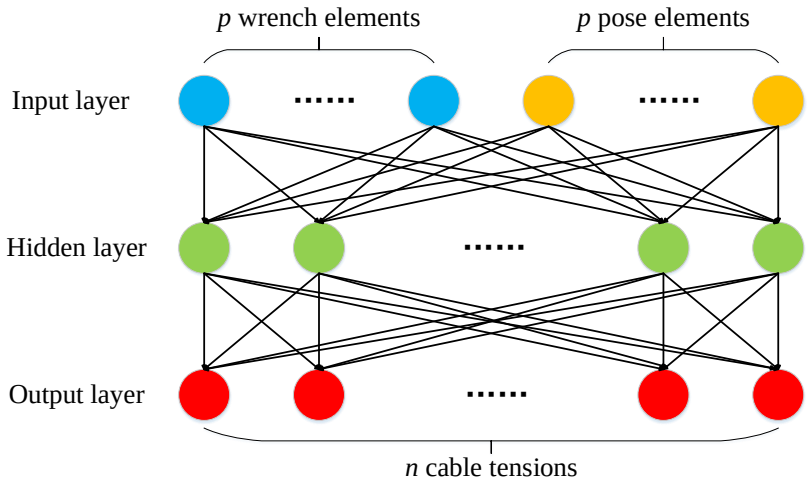


Figure 3.4 The architecture of the FNN used in the PID–FNN control strategy

The complexity problem of the FNN (i.e., how many units are necessary for the hidden layer to yield a certain degree of approximation) has been discussed in (Z. Chen et al., 2015). The required number of the hidden layer units of the FNN depends on the CDPR and the required degree of approximation. A larger number of hidden layer units of the FNN is necessary if the CDPR has more DOFs or the required degree of approximation is higher. Meanwhile, more data samples are usually needed to train the FNN with more hidden layer units to achieve the same degree of approximation. The more data samples required, the longer time it takes to collect them. Given a certain number of data samples, the number of hidden layer units of an FNN to approximate the data samples can be selected using the cross-validation approach (Haykin, Haykin, Haykin, Elektroingenieur, & Haykin, 2009).

The implementation of the PID–FNN control strategy is shown in Figure 3.5. The implementation has two parts, namely, training of the FNN and control of the CDPR. Training the FNN is to have the FNN learn how to map wrenches at a pose in the task space to sets of cable tensions in the joint space when the Jacobian of the CDPR is unknown. As shown at the top of Figure 3.5, to collect

data samples of wrenches, poses, and cable tensions to train the FNN, random cable tensions are applied to the CDPR first. The applied cable tensions generate a wrench which tends to move the CDPR. The applied cables tensions, the current pose, and the wrench generated by the applied cable tensions are recorded as one data sample. If the wrench cannot be measured directly, one can record the motion (i.e., pose, velocity, and acceleration) of the CDPR alternatively. The wrench generated by cable tensions can then be calculated based on the dynamic model of the CDPR. The pseudo-code to collect data and train the FNN used in the PID–FNN control strategy is shown in Algorithm 1.

The FNN in the PID–FNN control strategy maps a wrench applied on the end-effector at a pose in the task space to a set of cable tensions in the joint space. For a fully-constrained CDPR, different cable tension vectors may generate the same wrench at a pose in the task space (Lim, Yang, Yeo, & Mustafa, 2011). As a result, data samples collected in the above way may have the same wrench $\boldsymbol{w}$ and pose $\boldsymbol{x}$ but different cable tension vectors $\boldsymbol{\tau}$. Nonetheless, the FNN cannot map one set of wrench $\boldsymbol{w}$ and pose $\boldsymbol{x}$ to different cable tension vectors $\boldsymbol{\tau}$. Thus, the FNN is unable to approximate data samples in the above-mentioned scenario of a fully-constrained CDPR. In order to avoid this shortcoming of the FNN, one can discard data samples with the same wrench and pose but different cable tension vectors when collecting data samples.

When a certain number of data samples is recorded, it is ready to train the FNN. Sophisticated ANN optimization algorithms, such as ADAM (Kingma & Ba, 2014) and SGD (Recht, Re, Wright, & Niu, 2011), can be used to train the FNN. After training, the FNN can be used together with the PID controller to control the CDPR. As shown in Figure 3.5, the PID controller controls the CDPR in the task space and calculates the target wrench needed to move the CDPR from the current pose to the target pose. Then the FNN maps the target wrench at the current pose to a set of target cable tensions which will be delivered by actuators.

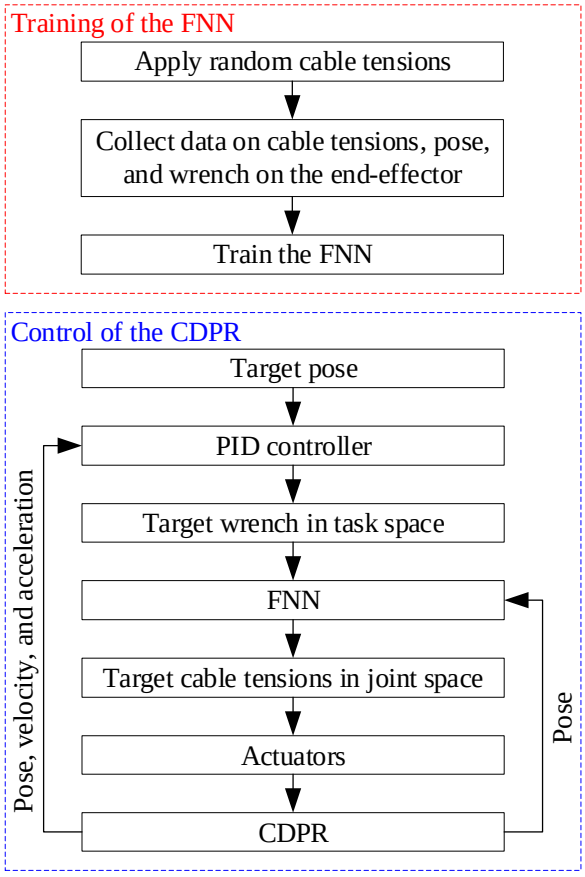


Figure 3.5 Flow diagram of the PID–FNN control strategy

Algorithm 1: Training the PID–FNN control strategy applied on a CDAD

```
Reset the CDAD
for the number of data samples do
    Randomly apply cable tensions
    Record the cable tensions and the pose
    Record the wrench calculated based on the dynamics model of the CDAD
    if the CDAD is out of the training workspace then
        Reset the CDAD
        Continue
    Stack the data sample (i.e., wrenches, poses, and cable tensions)
end for
Update the FNN based on the stack of data samples
```

Similar to the control strategy in (Khosravi & Taghirad, 2014b), the control of a CDPR via the PID–FNN control strategy in the task space is based on the dynamic model of the CDPR. The major difference of the PID–FNN control strategy from the control strategy in (Khosravi & Taghirad, 2014b) is the way to derive the set of cable tensions from a target wrench. Since the Jacobians are known in (Khosravi & Taghirad, 2014b), the set of cable tensions to generate the target wrench is simply calculated by solving the inverse dynamics problem. However, based on the PID–FNN control strategy, the set of cable tensions is the output of the trained FNN whose inputs are the target wrench and the current pose. Thus, even if the Jacobian of the CDPR is unknown, one still can use the PID–FNN control strategy to control the CDPR.

Assuming that the disturbance wrench caused by model uncertainties is bounded, it was proved in (Khosravi & Taghirad, 2014b) that a PID controller of a CDPR in the task space is stable of the form of uniformly ultimately bounded (UUB) with proper controller gains. The proper controller gains can be achieved by increasing

$$\beta = \min\{\alpha k_I, \alpha(k_P - k_D) - k_I, k_D\} \tag{3.23}$$

till the stability conditions are satisfied (Khosravi & Taghirad, 2014b). α is a scalar satisfied $0 < \alpha < 0.5$. k_P, k_I, and k_D are the gains of the PID controller. It is assumed that both disturbance wrenches (i.e., $\boldsymbol{w}_d$ caused by model uncertainties and $\boldsymbol{w}_N$ by the approximation error of the FNN)

are bounded. Therefore, according to (Khosravi & Taghirad, 2014b), the PID–FNN control strategy for a CDPR is also stable of the form of UUB.

3.2.2 Learning Control Strategies Based on Deep Reinforcement Learning

DRL introduces DNNs to solve reinforcement learning problems. With the help of DNNs, DRL can extract features from high-dimensional data and learn complex policies. DRL is especially suitable for sequential decision-making applications (e.g., robot manipulation) (Tsurumine et al., 2019). Many DRL algorithms have recently been developed into end-to-end DRL strategies used for a variety of robot manipulation tasks (F. Li et al., 2019; Z. Li et al., 2018; A Nagabandi et al., 2018; Passalis & Tefas, 2019; Rahman et al., 2018; M. Zhang et al., 2017; T. Zhang et al., 2016). An off-policy DRL strategy based on DNNs and guided policy search (GPS) was used to control a quadrotor to avoid obstacles in (T. Zhang et al., 2016). The off-policy DRL strategy employs the pose and the velocity of the quadrotor and data from 30 laser rangefinders installed on the quadrotor as its states. High-dimensional states are challenging to be integrated into non-learning-based control strategies, such as PID controllers, but can be easily processed by DRL strategies. In (A Nagabandi et al., 2018), TRPO was used to control bio-inspired robots in the MuJoCo simulator (Todorov et al., 2012). MDGPS was applied to the locomotion control of a tensegrity robot with 36 states in (M. Zhang et al., 2017). The high-dimension of states makes it hard to control the tensegrity robot using non-learning-based control strategies. Moreover, in (Rahman et al., 2018), a DQN was applied to control a self-balancing robot with two wheels in the Gazebo simulator (Koenig & Howard, 2004). When DRL strategies were used to control the robots in (F. Li et al., 2019; Z. Li et al., 2018; A Nagabandi et al., 2018; Passalis & Tefas, 2019; Rahman et al., 2018; M. Zhang et al., 2017; T. Zhang et al., 2016), the tasks of controlling the robots were treated as black boxes, without any knowledge of the internal workings of the robots. Therefore, DRL strategies are regarded as end-to-end DRL strategies (Levine, Finn, Darrell, & Abbeel, 2016).

The task of controlling a robot is not always a complete black box in practice. Besides inputs and outputs of the black box, one usually has certain knowledge about its internal workings. For example, when manipulating a robot, the robot manipulation task can be decomposed into multiple subtasks (Kober, Bagnell, & Peters, 2013). If one knows the internal workings of a subtask, this subtask may be accomplished by a non-learning-based approach (e.g. an inverse dynamics

equation or a PID algorithm). This raises the question of whether there is any benefit of integrating DRL with non-learning-based approaches in a robot manipulation task. To the best knowledge of the authors, the effects of integrating DRL with non-learning-based approaches on the learning speed and the robustness of DRL to model uncertainties have not been discussed in the literature.

To study the effects of integrating DRL with non-learning-based approaches on the learning speed and the robustness of DRL to model uncertainties, an end-to-end DRL strategy and a hybrid DRL strategy are developed and compared in this study in controlling a CDPR. The end-to-end DRL strategy, called the end-to-end DDPG strategy, is developed based on a DDPG algorithm. When the end-to-end DDPG strategy is applied, the task of controlling the CDPR is accomplished entirely by the DDPG algorithm. The hybrid DRL strategy, called the hybrid DDPG strategy, is developed by integrating a DDPG algorithm and the inverse dynamics equation of the CDPR. When the hybrid DDPG strategy is applied, some subtasks of the task of controlling the CDPR are accomplished by the DDPG algorithm, while the other subtasks are accomplished by the inverse dynamics equation of the CDPR.

3.2.2.1 A Learning Control Strategy Based on Deep Reinforcement Learning

Figure 3.6a illustrates how a CDPR is controlled to move to a target pose. Given the target pose, the controller of the CDPR needs to calculate a set of target cable tensions. Actuators are used to physically deliver the set of target cable tensions to drive the CDPR to the target pose. The task of the controller of the CDPR, calculating a set of target cable tensions from a given target pose, can be divided into two subtasks. The first subtask is to calculate the target wrench in the task space to drive the CDPR to the target pose. Since cables can be pulled only, the target wrench in the task space has to be converted to a set of target cable tensions in the joint space, which is the second subtask. Only when such a set of target cable tensions is available can actuators reel cables in or out to drive the CDPR to the target pose. In this section, a hybrid DDPG strategy and an end-to-end DDPG strategy are developed to control the CDPR. The hybrid DDPG strategy accomplishes the first subtask by DDPG and the second subtask by the inverse dynamics equation of the CDPR. The end-to-end DDPG strategy accomplishes the whole task (i.e., both the first and the second subtasks) of the controller of the CDPR using DDPG.

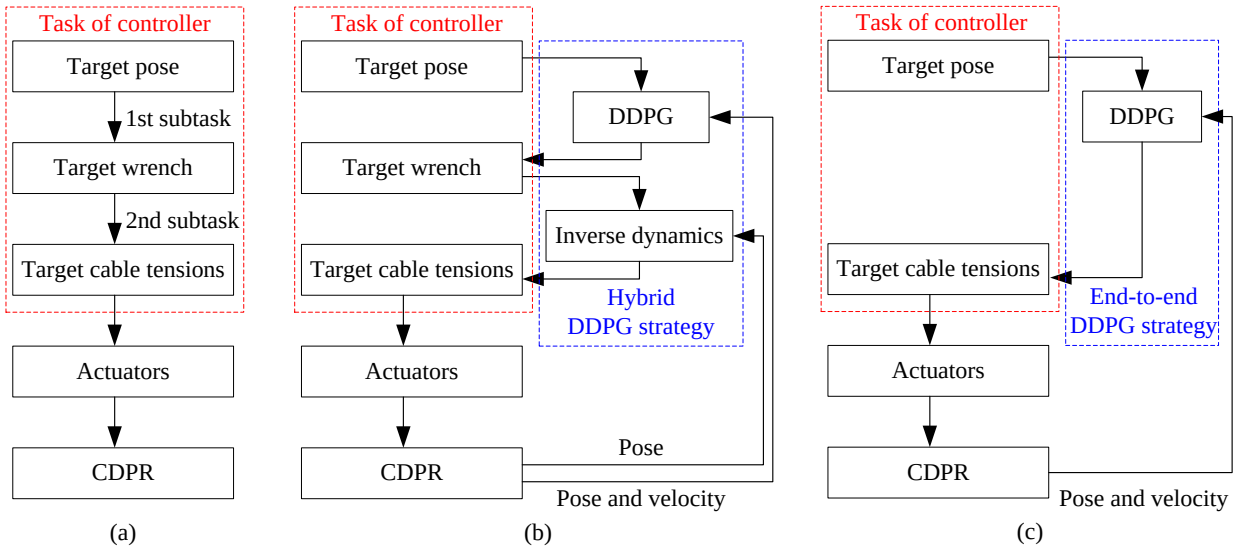


Figure 3.6 Flow diagrams of the DDPG-based strategies:

(a) the task of the controller of a CDPR in controlling the CDPR to a target pose; (b) flow diagram of the hybrid DDPG strategy; and (c) flow diagram of the end-to-end DDPG strategy

The flow diagram of the hybrid DDPG strategy is shown in Figure 3.6b. The hybrid DDPG strategy integrates DDPG and the inverse dynamics equation of a CDPR. The hybrid DDPG strategy accomplishes the first subtask of the controller of the CDPR by DDPG. To accomplish the first subtask, the DDPG in the hybrid DDPG strategy employs the following state variables: the pose of the CDPR at a certain time step, the difference between the target pose and the pose of the CDPR at a certain time step, and the velocity of the CDPR at a certain time step. Action variables of the DDPG in the hybrid DDPG strategy is the target wrench in the task space that is expected to drive the CDPR to the target pose. Once the first subtask is accomplished, the DDPG of the hybrid DDPG strategy outputs the target wrench in the task space. The second subtask takes such a target wrench in the task space and converts it to a set of target cable tensions. The second subtask is accomplished by solving the inverse dynamics equation of the CDPR. When converting the target wrench in the task space to a set of target cable tensions of a fully-constrained CDPR by solving the inverse dynamics equation of the CDPR, there is an infinite number of feasible tension distributions. The optimal tension distribution is calculated using the pseudo-inverse of Jacobian (Babaghasabha et al., 2015). Once the optimal tension distribution is obtained, actuators physically deliver the set of target cable tensions. Algorithm 2 illustrates the implementation of the hybrid DDPG strategy.

The flow diagram of the end-to-end DDPG strategy is shown in Figure 3.6c. The end-to-end DDPG strategy accomplishes the whole task (i.e., both the first and the second subtasks) of the controller of the CDPR using DDPG. State variables of the DDPG in this strategy consist of the pose of the CDPR at a certain time step, the difference between the target pose and the pose of the CDPR at a certain time step, and the velocity of the CDPR at a certain time step as well. However, action variables of the DDPG in this strategy is a set of target cable tensions in the joint space, rather than a target wrench in the task space. The set of target cable tensions is derived based on the behavior policy of the DDPG. Algorithm 3 illustrates the implementation of the end-to-end DDPG strategy.

Algorithm 2: Implementation of the hybrid DDPG strategy in controlling a CDPR

Randomly initialize the actor network and the critic network
Initialize the actor-target network and the critic-target network
Initialize an experience replay buffer
for number of episodes **do**
    Reset the CDPR
    Randomly set a target pose
    Record the $\boldsymbol{s}_t$
    for maximum number of time steps **do**
        Calculate $\boldsymbol{a}_t$ (target wrench) according to the current policy and exploration noise
        Calculate the optimal tension distribution based on the inverse dynamics equation
        Execute the optimal tension distribution and observe reward and $\boldsymbol{s}_{t+1}$
        if the CDPR is out of the training workspace **then**
            Break;
        if the CDPR reaches the target pose **then**
            Break;
        Stack the data (i.e., states, actions, reward, and next states) in the replay buffer
        Select a mini-batch of data from the replay buffer
        Update the critic network via minimizing the training loss
        Update the actor network using the sampled policy gradient
        Update the actor- and critic-target networks
    end for
end for

Algorithm 3: Implementation of the end-to-end DDPG strategy in controlling a CDPR

Randomly initialize the actor network and the critic network
Initialize the actor-target network and the critic-target network
Initialize an experience replay buffer
for number of episodes **do**
    Reset the CDPR
    Randomly set a target pose
    Record the $\boldsymbol{s}_t$
    for maximum number of time steps **do**
        Calculate $\boldsymbol{a}_t$ according to the current policy and exploration noise
        Execute $\boldsymbol{a}_t$and observe reward and $\boldsymbol{s}_{t+1}$
        if the CDPR is out of the training workspace **then**
            Break;
        if the CDPR reaches the target pose **then**
            Break;
        Stack the data (i.e., states, actions, reward, and next states) in the replay buffer
        Select a mini-batch of data from the experience replay buffer
        Update the critic network via minimizing the training loss
        Update the actor network using the sampled policy gradient
        Update the actor- and critic-target networks
        end for
end for

3.2.3 Safety Robustness of Learning Control Strategies

Reinforcement learning (RL) studies how an agent learns a policy by maximizing the expected cumulative rewards when interacting with the environment (Henderson et al., 2018). In recent years, RL has achieved dramatic success in many areas such as robot manipulation (Lillicrap et al., 2015), playing Go (Silver et al., 2016), and playing Atari (Mnih et al., 2013). Although the success of RL has made a deep impression to the control community, the RL community and the control community still remain practically disjointed (Recht, 2018). On the other hand, solving problems in safety-critical applications, such as self-driving of vehicles and robot manipulation,

requires the deep fusion of both machine learning and control technologies (Recht, 2018). For safety-critical applications, the failure of a learned policy may lead to dangerous situations (Pattanaik, Tang, Liu, Bommannan, & Chowdhary, 2018). Appropriate control techniques from the control community can be used to improve the safety of learned policies (Recht, 2018).

Reward functions may not always be specified for some RL problems (e.g., robot manipulation) (Y. Li, 2017). Reward shaping (Ng, Harada, & Russell, n.d.) is an inevitable and significant step to solve these problems. In reward shaping, one may design various reward functions to test how to better guide an agent to learn. Different policies can be learned by the agent with different reward functions. The robustness of the learned policies (called policies for short) is usually evaluated based on either the achieved reward (Z. Gu, Jia, & Choset, n.d.; Pattanaik et al., 2018; Pinto, Davidson, Sukthankar, & Gupta, 2017) or the success rate (Peng, Andrychowicz, Zaremba, & Abbeel, 2018; Tobin et al., 2017). When using RL to solve problems in safety-critical applications, the robustness of policies needs to be evaluated from the perspective of safety as well. However, a method to evaluate the robustness of policies from the perspective of safety is not available currently.

Robustness plays a pivotal role since its introduction in the 1860s in the control community (Bhattacharyya, 2017). Robust control studies from robustness through quadratic optimization to robustness under parametric uncertainty in recent years (Bhattacharyya, 2017). Robust control aims to minimize the effects of unknown initial conditions and external influences on system behavior, subject to the constraints of not having a complete representation of the system (Dullerud & Paganini, 2013). Achieving close loop stability and tracking performance while providing adequate stability margins are the main goal of robust control (Eugene, Kevin, & Howe, 2013). A robust controller can be thought of as a policy capable of regulating a system whose dynamics may contain bounded uncertainties (Eugene et al., 2013).

The robustness of a policy is the policy's ability to maintain functionality when there are uncertainties and perturbations (Bhattacharyya, 2017). Various types of robustness are summarized in Table 3.1. Reward robustness of a policy refers to the policy's ability to achieve a high reward (Z. Gu et al., n.d.; Pattanaik et al., 2018; Pinto et al., 2017). Success robustness of a

policy refers to the policy's ability to successfully solve the RL problem (Peng et al., 2018; Tobin et al., 2017). When it comes to robust control, stability robustness (Yedavalli, 1985) of a policy (i.e., a controller) refers to the policy's ability to stabilize the state. Reward robustness and success robustness have been intensively discussed in the RL community while stability robustness has been studied in the control community.

Table 3.1 Robustness of policies

Type of Robustness	**Description**
Reward robustness	Ability of a policy to achieve a high reward
Success robustness	Ability of a policy to solve the RL problem
Stability robustness	Ability of a policy to stabilize state
Safety robustness	Ability of a policy to maintain safe state

The robustness proposed in this study is called safety robustness. Safety robustness refers to a policy's ability to maintain safe state. Since safe state does not necessary mean that the RL problem is successfully solved, safety robustness is generally different from success robustness. Moreover, safety robustness is different from stability robustness as well because to maintain the state within a safe range is not necessary to stabilize the state to a certain state. The safe range is usually bounded in practice, while the state of an unstable system is usually diverging and exceeding the safe range. Therefore, an unstable system is usually unsafe. This suggests that the stability robustness of a policy affects the safety robustness of the policy.

Robot manipulation can be treated as an RL problem. Such an RL problem often bears safety concerns. Various RL algorithms, such as DDPG (Lillicrap et al., 2015), DQN (Mnih et al., 2013), TRPO (Schulman et al., 2015), MDGPS (Montgomery & Levine, 2016), Normalized Advantage Function (NAF) (S. Gu, Lillicrap, Sutskever, & Levine, 2016), and Proximal Policy Optimization (PPO) (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017), have been used to solve a variety of robot manipulation problems (F. Li et al., 2019; Z. Li et al., 2018; Lowrey, Kolev, Dao, Rajeswaran, & Todorov, 2018; A Nagabandi et al., 2018; Passalis & Tefas, 2019; Rahman et al., 2018; Tan et al., 2018; Xie, Berseth, Clary, Hurst, & Panne, 2018; M. Zhang et al., 2017; T. Zhang et al., 2016). Duan et al. (Duan, Chen, Houthooft, Schulman, & Abbeel, 2016) provided baseline implementations for developing RL algorithms for robot manipulation. Various methods (e.g.,

injecting noise (Jakobi, Husbands, & Harvey, 1995), robust RL (Z. Gu et al., n.d.; Pattanaik et al., 2018; Pinto et al., 2017), domain randomization (Tobin et al., 2017), and dynamics randomization (Peng et al., 2018)) have been used to improve the reward robustness and the success robustness of policies for robot manipulation. However, safety robustness of policies has not been discussed for robot manipulation.

Researchers have discussed safe RL (Garcia & Fernández, 2012; Garcıa & Fernández, 2015), a process of learning policies that maximize the expectation of reward in problems in which it is important to respect safety constraints during the learning and deployment process. Moreover, risk-sensitive RL (Geibel & Wysotzki, 2005) has been discussed to find less risky policies. The robust Markov Decision Process (MDP) has been discussed to minimize the cost when the parameters have uncertainties (Osogami, 2012). However, the above-mentioned studies (Garcia & Fernández, 2012; Garcıa & Fernández, 2015; Geibel & Wysotzki, 2005; Osogami, 2012) aimed to find safe policies, rather than to evaluate how safe the policies are when subject to uncertainties and perturbations.

3.2.3.1 Stability Margin of Policies for Solving an LQR Problem

A Linear Quadratic Regulator (LQR) problem is a class of optimal control problems with convex quadratic cost functions and linear dynamics. An LQR problem can be mathematically described as (Recht, 2018)

$$\text{minimize } E\left(\frac{1}{2}\sum_{t=0}^{N} x_t^T \boldsymbol{Q} x_t + u_t^T \boldsymbol{R} u_t + \frac{1}{2} x_{N+1}^T \boldsymbol{S} x_{N+1}\right) \tag{3.24}$$

$$\text{subject to } x_{t+1} = \boldsymbol{A} x_t + \boldsymbol{B} u_t,\ u_t = \pi_t(x_t) \tag{3.25}$$

where $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{S}$ are positive semi-definite matrices. x_t is the state at step *t* and u_t is the action at step *t*. The state transition is governed by the update rule with matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The quadratic cost function of the LQR problem can be regarded as a negative reward function from the perspective of RL. It has been shown that RL algorithms can guide an agent to learn the optimal policy to solve an LQR problem (Recht, 2018).

Since the stability robustness of a policy affects the safety robustness of the policy and control techniques in the control community are available to analyze the stability robustness of a policy, the author studies the stability robustness of the optimal policy before discussing safety robustness.

The stability robustness of a system (e.g., a system that has a policy and an environment) to uncertainties or perturbations can be evaluated by Structured Singular Value (SSV) (Doyle, Wall, & Stein, 1982). SSV is a scalar index. A larger SSV means the system has a stronger ability to stabilize state when subject to uncertainties or perturbations. Consider a discrete-time double integrator system with the dynamic model

$$x_{t+1} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix} u_t \tag{3.26}$$

with the cost function defined as

$$J = \sum_{t=0}^{N} (x_t^T \begin{bmatrix} 1 & 0 \\ 0 & q_{22} \end{bmatrix} x_t + u_t^T r_0 u_t) \tag{3.27}$$

In this case, matrices $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{S}$ in (3.24) are

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 \\ 0 & q_{22} \end{bmatrix}, \boldsymbol{R} = r_0, \boldsymbol{S} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.28}$$

where q_{22} ranges from 0 to 0.5 and r_0 from $1e-6$ to $1e-1$. q_{22} is the penalty of the state changing rate. r_0 is the penalty of action. Assume the state has at most 20% error when it is observed by an agent, and the action has at most 20% error when it is applied to the environment as well. Then, the SSV of the system controlled by optimal policies corresponding to the cost functions with variables q_{22} and r_0 can be calculated (Doyle et al., 1982).

As shown in Figure 3.7, the SSV, denoted by μ, of the system controlled by optimal policies corresponding to different cost functions are generally different. It means that the stability robustness of the above-mentioned optimal policies is generally different. The stability robustness of the above-mentioned optimal policies is indeed affected by the cost functions. This suggests that, for an LQR problem, if an RL algorithm guides an agent to learn an optimal policy as shown in (Recht, 2018), the stability robustness of the optimal policy to uncertainties and perturbations is generally affected by the reward function. As discussed in section 3.2.3.1, the stability robustness of a policy affects the safety robustness of the policy. Hence, the safety robustness of a policy to uncertainties and perturbations is generally affected by the reward function as well. This motivated the authors to further study the safety robustness of policies, since reward functions may not always be specified for some RL problems (Y. Li, 2017).

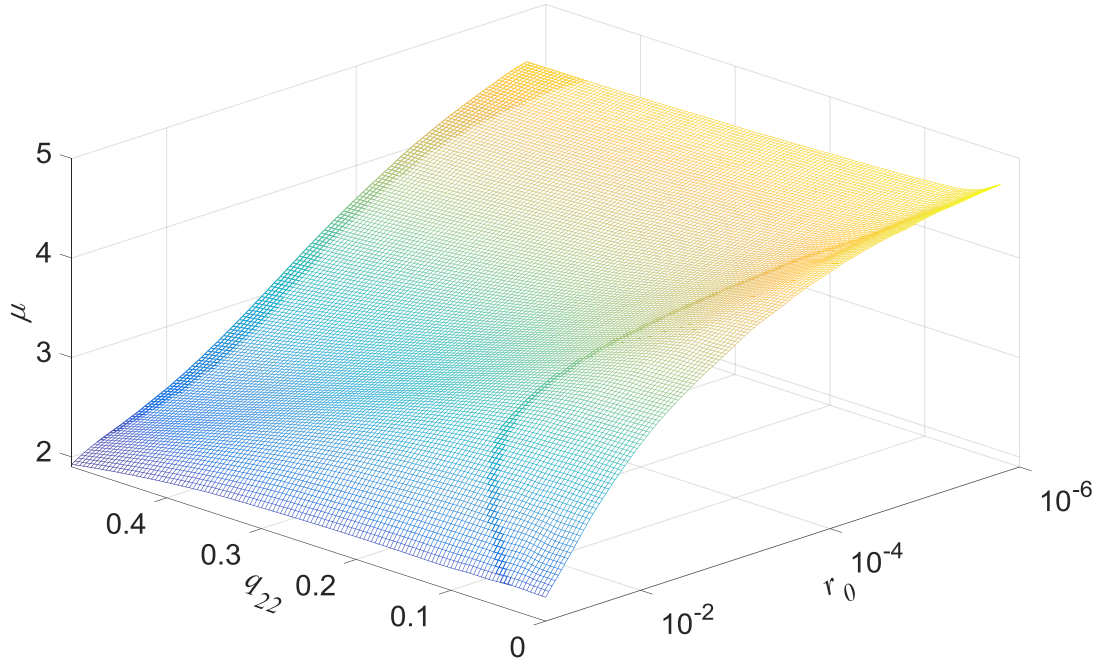


Figure 3.7 The SSV of the system controlled by optimal policies

3.2.3.2 A Method to Approximate Safety Robustness of Learning Control Strategies

To evaluate the robustness of policies of an RL problem from the perspective of safety, the system architecture of robust control (Eugene et al., 2013) is applied to the RL problem, as shown in Figure 3.8. To adapt to the RL problem, the controller and the plant of the original system architecture of robust control are replaced by the policy and the environment, respectively. Δ_1 represents the uncertainty of action and Δ_2 represents the uncertainty of state. According to (Eugene et al., 2013), Δ_1 and Δ_2 can be constructed to model any type of uncertainties. More specifically, Δ_1 can be used to model uncertainties from actuators, dynamics, time delays, or any environment inputs, while Δ_2 can be used to model uncertainties from sensors, dynamics, time delays, or any environment outputs. Since uncertainties are applied to the state and the action of the RL problem, rather than the RL algorithm, the reward function, or the environment of the RL problem, the system architecture in Figure 3.8 is generally applicable to any RL problems to study the robustness of policies.

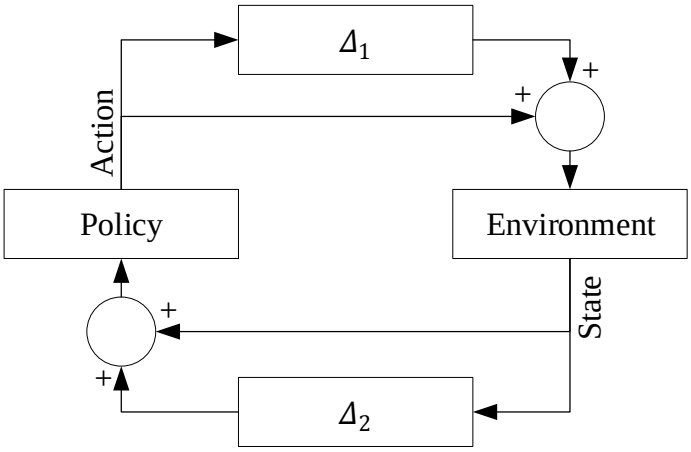


Figure 3.8 The system architecture of robust control applied to an RL problem

Based on the system architecture in Figure 3.8, one can approximate the safety robustness of policies via applying the policies to solve an RL problem when subject to Δ_1 and Δ_2. The pseudocode for approximating the safety robustness of policies is illustrated in Algorithm 4. To approximate the safety robustness of policies, one needs to determine the number of policies L, the safe states, the maximum magnitude of uncertainties of action Δ_a^{limit}, the maximum magnitude of uncertainties of state Δ_s^{limit}, the number of episodes M, and the number of steps of an episode T. A counter, $k_l(\Delta_s^{max}, \Delta_a^{max})$, is used to count the number of safe episodes when a policy is subjected to Δ_s^{max} and Δ_a^{max}.

Algorithm 4: Approximate Safety Robustness of Policies

Load all L policies
Define safe states
for $\Delta_s^{max} = 0, \Delta_s^{limit}$ **do**
  for $\Delta_a^{max} = 0, \Delta_a^{limit}$ **do**
    Randomize Δ_a and Δ_s, satisfying $|\Delta_a| \leq \Delta_a^{max}$ and $|\Delta_s| \leq \Delta_s^{max}$
    for $l = 1, L$ **do**
      $k_l(\Delta_s^{max}, \Delta_a^{max}) = 0$
    end for
    for episode = 1, M **do**
      for $l = 1, L$ **do**
        Initialize the environment
        Randomize a target if necessary
        for step = 1, T **do**
          Apply π_l with Δ_a and Δ_s
        end for
        if all states when applying π_l are within the safe ranges **then**
          $k_l(\Delta_s^{max}, \Delta_a^{max}) = k_l(\Delta_s^{max}, \Delta_a^{max}) + 1$
      end for
    end for
  end for
end for
The safety robustness of π_l can be approximated $\frac{k_l(\Delta_s^{max}, \Delta_a^{max})}{M}$

3.3 Verify the Learning Control Strategies in Simulations

To validate the fundamental studies of CDADs, Matlab 2015b is used to evaluate workspace isotropy and to analyze joint force. To study the learning control strategies for controlling CDADs, Gazebo 7.0 (Koenig & Howard, 2004) and ROS kinetic 1.12.14 are used to simulate the dynamics

of CDADs and TensorFlow 1.12.0 and Keras 1.0.5 are used to build neural networks based on Python 2.7.

Training a strategy on a real CDAD is possible to damage the CDAD, especially when the strategy has not been well-trained in the initial stage. Thus, a CDAD model in a simulator, rather than a real CDAD, is highly desirable as simulations are always safe for the real robot (Kober et al., 2013). Moreover, a simulator is also helpful in developing and fine-tuning DRL algorithms because the iteration of simulations is much faster than that of real experiments (Kober et al., 2013). Therefore, simulations will be conducted to verify the effectiveness of learning control strategies. The model of the CDAD is established in the Gazebo simulator as shown in Figure 3.9. Cable tensions will be applied between every pair of anchor points of the CDAD.

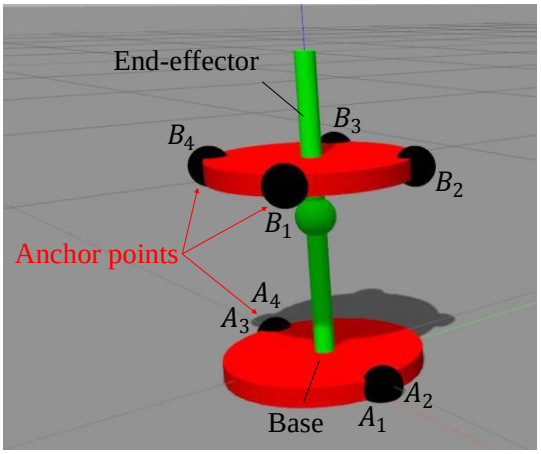


Figure 3.9 The model of the example CDAD in the Gazebo Simulator

3.4 Summary

For chapter three, fundamental studies of CDADs – workspace isotropy and joint force analysis - are presented. A PID–FNN control strategy and DDPG-based strategies are proposed for CDADs to control the pose of the human joint. Besides, simulations are designed to validate the learning control strategies. Simulations of research methodologies proposed in chapter three will be introduced in chapter four.

CHAPTER 4. RESULTS

According to the research methodologies discussed in chapter three, simulations are conducted to validate the proposed concepts (i.e. workspace isotropy, moment efficiency, and safety robustness) and strategies (a PID–FNN control strategy and DDPG-based strategies) in chapter four. Sections 4.1.1, 4.1.2, 4.2, and 4.3 of chapter four are revised based on the author's published papers (Xiong & Diao, 2018a), (Xiong et al., 2018), (Xiong, Zhang, et al., 2019), and (Xiong, Ma, et al., 2019), respectively.

4.1 Fundamental Studies of Cable-Driven Assistive Devices

4.1.1 Evaluation of Workspace Isotropy Indices

To assess the effectiveness of TWII, RWII, and EWII in reflecting the robustness of a CDPR to rotational disturbances to its movable base, random rotations are applied to the bases of a CDPR in simulation. TWII, RWII, and EWII are assessed on the CDPR. Example designs of a planar CDPR with three DOFs and four cables are used in the simulation.

4.1.1.1 Simulation Setup

The simulation is conducted in MATLAB on a computer with an Intel Core i7-6700 CPU @ 3.40 GHz and 16 GB of RAM. The time to compute the proposed workspace isotropy indices is affected by the interval size in simulation, the workspace determination method, etc. Smaller interval size can improve the accuracy of the proposed indices, but it also increases the computational cost at the same time.

The effectiveness of workspace isotropy indices is assessed with the planar CDPR shown in Figure 4.1. The four anchor points $A_1, A_2, A_3,$ and A_4 on the base forms a square depicted with dashed lines. The red lines connecting anchor points $B_1, B_2, B_3,$ and B_4 depict the end-effector. The three example designs of the CDPR have the same base, but different end-effectors. The end-effectors of the three example designs are square, vertical rectangular, and horizontal rectangular, respectively. The origin of the base frame *xy* is located at the centroid of the square base and it is also the IAC. The origin of the end-effector frame is located at the centroid of the end-effector.

The pose of the CDPR is represented by (x, y, θ) (unit: mm, mm, deg). When the base frame coincides with the end-effector frame, coordinates of the anchor points on the base and the end-effectors of all three designs are listed in Table 4.1.

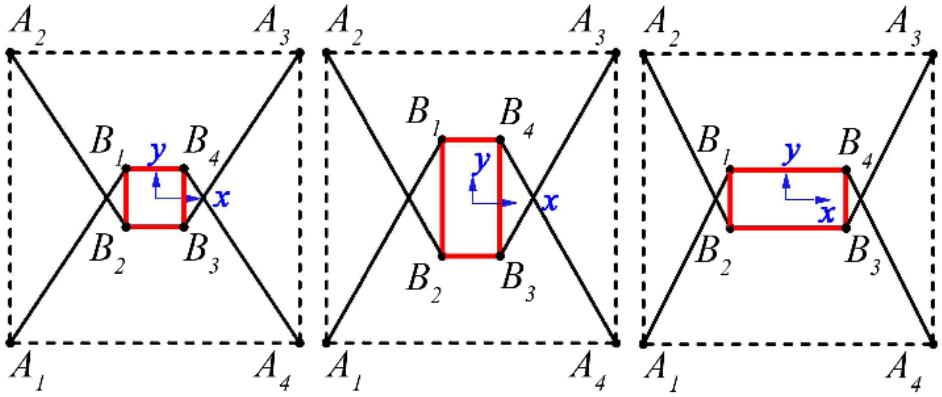


Figure 4.1 Three example designs of a planar CDPR:
(a) square design; (b) vertical rectangular design; and (c) horizontal rectangular design

Table 4.1 Positions of the anchor points (unit: mm)

Base in Figure 4.1	**End-effector in Figure 4.1a**	**End-effector in Figure 4.1b**	**End-effector in Figure 4.1c**
A_1: (-250, -250)	B_1: (-50, 50)	B_1: (-50, 100)	B_1: (-100, 50)
A_2: (-250, 250)	B_2: (-50, -50)	B_2: (-50, -100)	B_2: (-100, -50)
A_3: (250, 250)	B_3: (50, -50)	B_3: (50, -100)	B_3: (100, -50)
A_4: (250, -250)	B_4: (50, 50)	B_4: (50, 100)	B_4: (100, 50)

The workspaces of the example designs of the CDPR need to be determined before one can assess their isotropy using the proposed workspace isotropy indices. Researchers have proposed quite a few approaches for workspace determination. The force- or wrench-closure workspaces of the example designs of the CDPR are determined using the workspace determination approach proposed in (Gallina & Rosati, 2002; Gouttefarde & Gosselin, 2004). The translational workspaces and the entire workspaces of the three example designs of the CDPR are shown in Figure 4.2 and Figure 4.3.

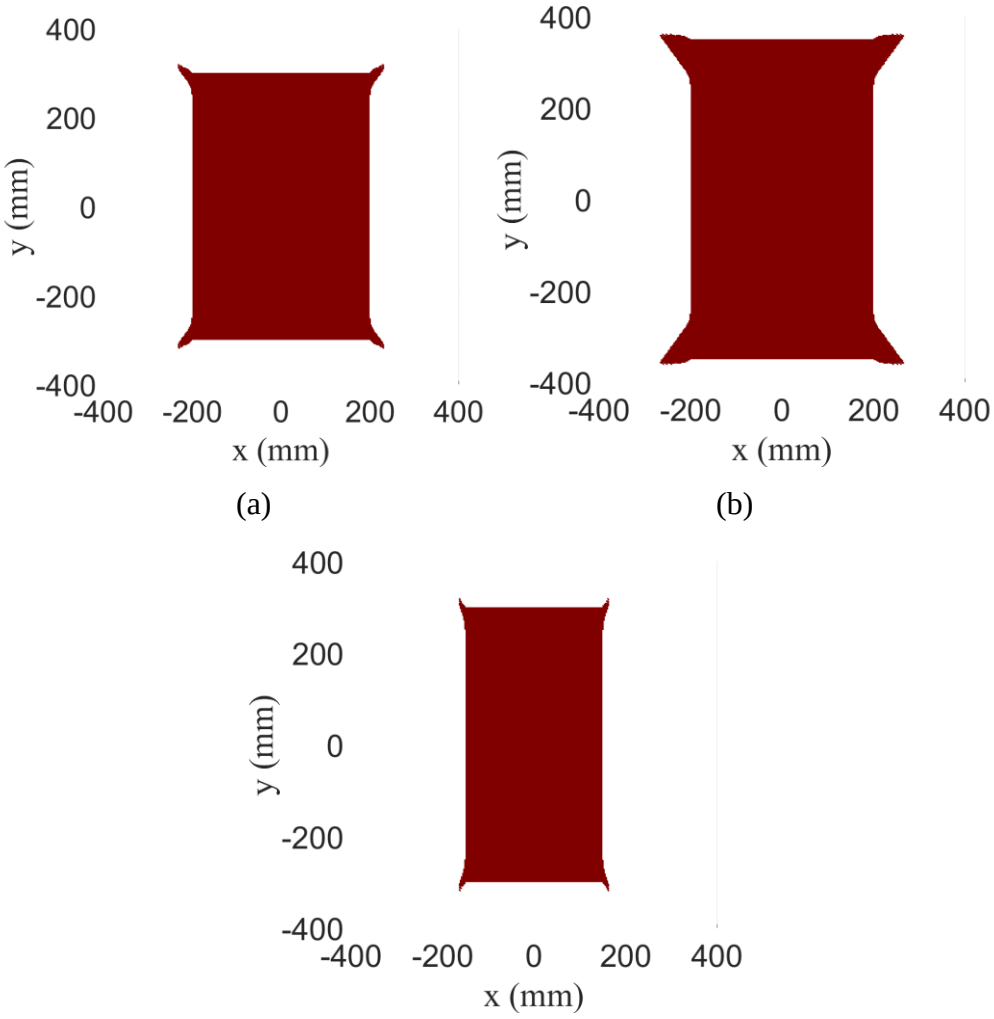
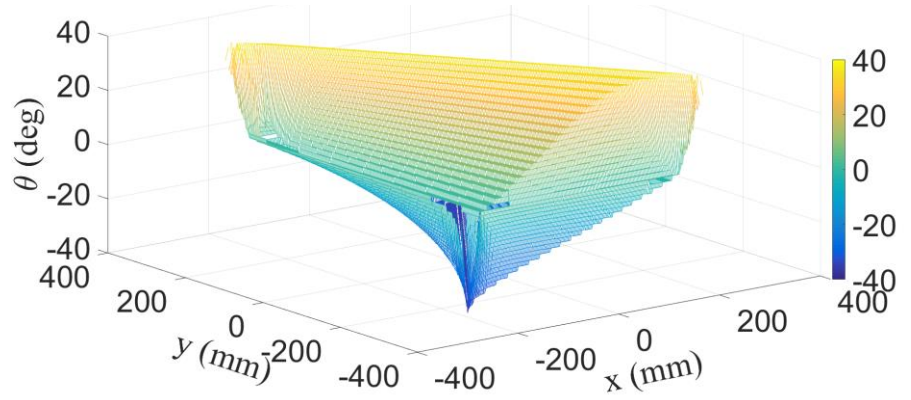
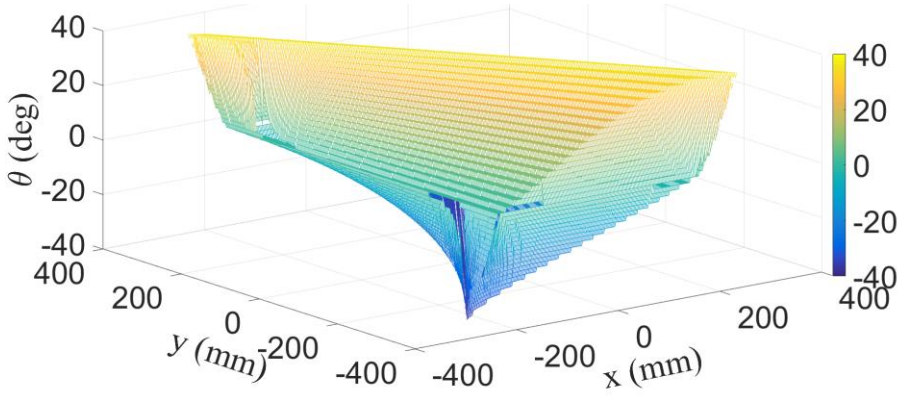


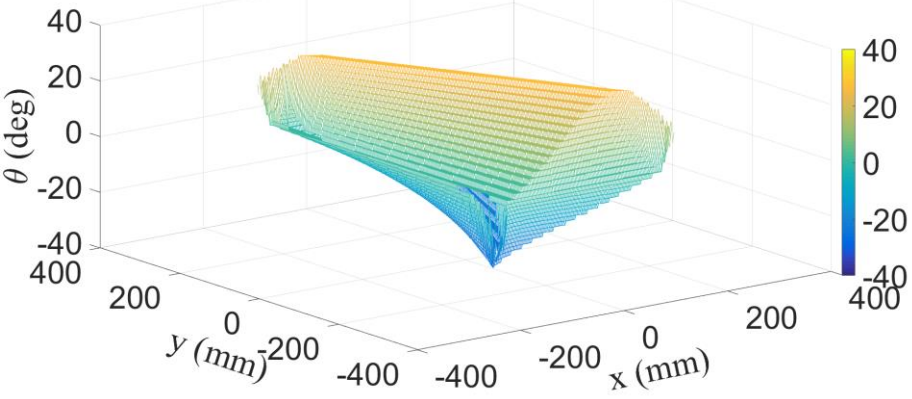
Figure 4.2 Translational workspaces of example designs of the planar CDPR: (a) square design; (b) vertical rectangular design; and (c) horizontal rectangular design



(a)



(b)



(c)

Figure 4.3 Entire workspaces of example designs of the planar CDPR:
(a) square design; (b) vertical rectangular design; and (c) horizontal rectangular design

4.1.1.2 Simulation Results

Workspace isotropy indices, TWII, RWII, and EWII, are applied to evaluate the workspace isotropy of the three example designs of the CDPR as shown in Figure 4.1. The calculated TWII, RWII, and EWII of the three example designs are listed in Table 4.2. It takes about 16 minutes to determine the workspace and compute all three workspace isotropy indices for one example design of the CDPR. The EWII of the planar CDPR is much smaller than RWII and TWII. It means that, after a random rotation of the base of the planar CDPR, the possibility for the three-dimensional entire workspace to overlap the original three-dimensional entire workspace is much smaller than the possibility for the two-dimensional translational workspace (or one-dimensional rotational workspace) to overlap the original two-dimensional translational workspace (or one-dimensional rotational workspace).

Table 4.2 TWII, RWII, and EWII of the three example designs of the CDPR

Example designs of the CDPR	**TWII**	**RWII**	**EWII**
Figure 4.1a	0.692	0.204	0.099
Figure 4.1b	0.659	0.204	0.100
Figure 4.1c	0.641	0.154	0.077

To verify the effectiveness of TWII, RWII, and EWII, 1,000 random rotations about the IAC, ranging from -180 degrees to 180 degrees, are applied to the movable bases of the three example designs of the CDPR in simulation. The simulation results are listed in Table 4.3, Table 4.4, and Table 4.5. It is shown that the percentage of the original translational workspace, rotational workspace, and entire workspace that can be reached by a CDPR match the corresponding TWII, RWII, and EWII, respectively. After a random rotation of the base, the design with higher TWII, RWII, and EWII can reach a higher percentage of its original translational workspace, rotational workspace, and entire workspace, respectively. Therefore, the robustness of the translational workspace, rotational workspace, and entire workspace of a CDPR to rotational disturbances to its movable base can be reflected by TWII, RWII and EWII, respectively.

Table 4.3 Percentage of the original translational workspace that can be reached

Example designs of the CDPR	**TWII**	**Percentage of the original translational workspace that can be reached**

Figure 4.1a	0.692	69.24%
Figure 4.1b	0.659	65.56%
Figure 4.1c	0.641	64.41%

Table 4.4 Percentage of the original rotational workspace that can be reached

Example designs of the CDPR	**RWII**	**Percentage of the original rotational workspace that can be reached**
Figure 4.1a	0.204	20.77%
Figure 4.1b	0.204	20.84%
Figure 4.1c	0.154	15.72%

Table 4.5 Percentage of the original entire workspace that can be reached

Example designs of the CDPR	**EWII**	**Percentage of the original entire workspace that can be reached**
Figure 4.1a	0.099	10.69%
Figure 4.1b	0.100	11.04%
Figure 4.1c	0.077	8.74%

4.1.2 Joint Force Analysis

One knee with one DOF (i.e., flexion and extension) assisted by an example three-DOF four-cable CDAD, as shown in Figure 4.4, is discussed based on the moment efficiency index in quasi-static conditions in this section.

4.1.2.1 Simulation Setup

The collision between cables, cuffs, the shank, and the upper leg are not taken into account in this discussion. One cuff with a radius of 0.1 m is worn on the upper leg and the shank, respectively. The shank is able to rotate about x_e axis. The angle of the knee, denoted as ϕ_k, is measured from the shank to z_b axis. Let d_u denote the distance between the centroid of the cuff worn on the upper leg and the knee joint. Let d_s denote the distance between the centroid of the cuff worn on the shank and the knee joint. Both d_u and d_s are variables in this study. Initially, F_e coincides with F_b, and the positions of the anchor points are shown in Table 4.6.

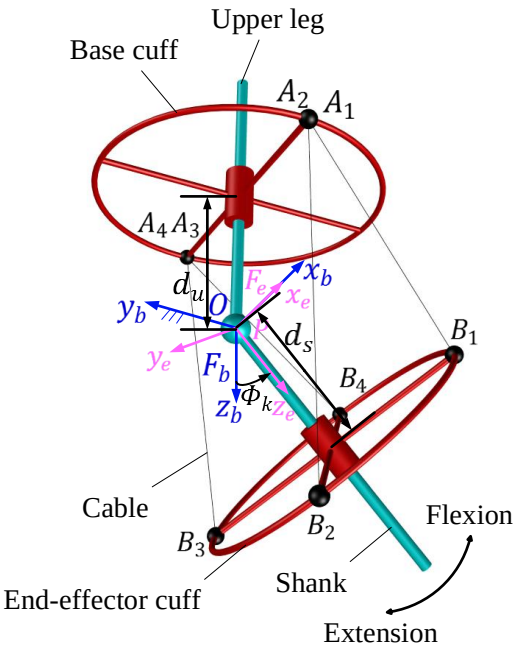


Figure 4.4 Notations of a knee assisted by a three-DOF four-cable CDAD

Table 4.6 Positions of anchor points (unit: m)

Anchor points	**Positions of Anchor points on the base (upper leg)**	**Positions of Anchor points on the end-effector (shank)**
1	A_1: $[0.1, 0.0, -d_u]^T$	B_1: $[0.0707, -0.0707, d_s]^T$
2	A_2: $[0.1, 0.0, -d_u]^T$	B_2: $[0.0707, 0.0707, d_s]^T$
3	A_3: $[-0.1, 0.0, -d_u]^T$	B_3: $[-0.0707, 0.0707, d_s]^T$
4	A_4: $[-0.1, 0.0, -d_u]^T$	B_4: $[-0.0707, -0.0707, d_s]^T$

4.1.2.2 Simulation Results

If the angle of the knee, the position of the upper leg cuff, and the position of the shank cuff are given, the maximum moment efficiency index, denoted by η_{max}, is achieved when the antagonistic cable tensions are all zero. In the simulation, the maximum moment efficiency index is calculated in the force-closure workspace of the CDAD. The maximum moment efficiency of the CDAD with respect to d_s and d_u in assisting the flexion and extension of the knee are plotted in Figure 4.5 and Figure 4.6, respectively.

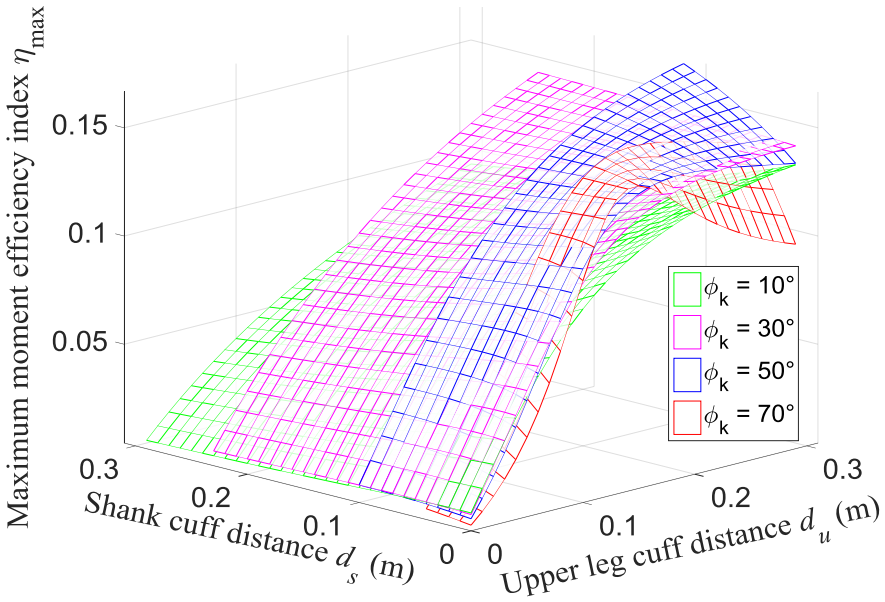


Figure 4.5 Maximum moment efficiency index of the CDAD assisting the flexion of knee

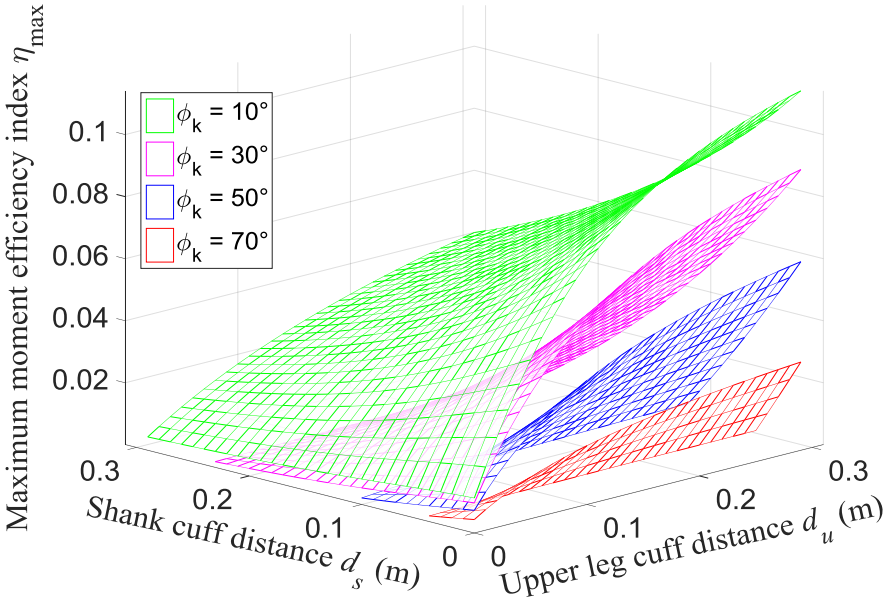


Figure 4.6 Maximum moment efficiency index of the CDAD assisting the extension of knee

According to Figure 4.5 and Figure 4.6, the angle of the knee and the distances from both cuffs to the knee joint significantly affect the maximum moment efficiency index. When assisting the knee in flexion, the maximum moment efficiency index does not change much with the shank cuff moving closer to the knee (i.e., d_s decreases) or the flexion of the knee (i.e., ϕ_k increases).

However, when the upper leg cuff is within 0.1 m from the knee ($d_u \leq 0.1$ m), the maximum moment efficiency index decreases sharply with the upper leg cuff moving closer to the knee (i.e., d_u decreases). When assisting the knee in extension, the maximum moment efficiency index increases with the shank cuff moving closer to the knee (i.e., d_s decreases), the upper leg cuff moving farther away from the knee (i.e., d_u increases), or the flexion of the knee (i.e., ϕ_k increases).

Based on Figure 4.5 and Figure 4.6, one can conclude that if the CDAD showed in Figure 4.4 is used for knee rehabilitation or assistance, wearing the shank cuff close to the knee allows the CDAD to assist the knee with a large angle of the knee while wearing the upper leg cuff far away from the knee (up to about 0.15 m) can improve the moment efficiency of the CDAD. Findings on positioning the cuffs provide a cornerstone for further study (e.g., control and assistance strategies) of this CDAD.

4.2 A Learning Control Strategy Based on a Feedforward Neural Network

The PID–FNN control strategy is applied to control the fully-constrained CDPR shown in Figure 4.7 in this section. The CDPR with three rotational DOFs and four cables is designed for the rehabilitation training of a human joint (e.g., ankle). Since a pose of the CDPR can be measured by an Inertial Measurement Unit (IMU) in practice, the pose of the CDPR is assumed to be known in the simulation. Even though the pose of the CDPR is known, due to the wearing inconsistency among training setups, the positions of anchor points in each training setup are unknown and thus, the Jacobian of the CDPR is unknown (Diao & Ma, 2008).

4.2.1 Simulation Setup

The feasible cable tension that can be delivered by an actuator of the CDPR is assumed to range from 0 to 10 N. In the simulation, a random disturbance wrench (i.e., $\boldsymbol{w}_d$) caused by model uncertainties is applied to the CDPR. The random disturbance is a 3-dimensional vector varying randomly over time. The direction of the random disturbance wrench distributes uniformly in 3-dimensional space. The magnitude of the random disturbance wrench satisfies a uniform distribution within a range from 0 to 0.1 Nm. The random disturbance wrench is applied both in collecting data samples and in controlling the CDPR. Namely, the collected data samples are with

noise and controlling the CDPR is with disturbance. Gravity is neglected in the simulation to simplify the analysis. F_b represents the base frame and F_e represents the end-effector frame. The attitude of F_e with respect to F_b is described by a vector of three Euler angles $[\phi, \theta, \psi]^T$ with a ψ-θ-ϕ sequence. The parameters of the CDPR are shown in Table 4.7.

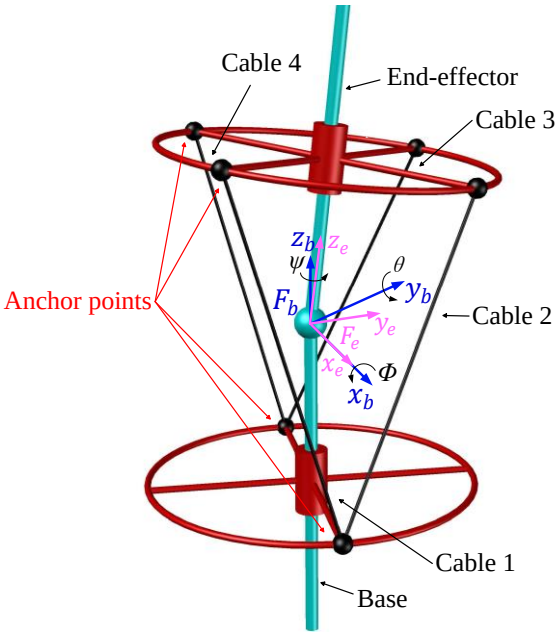


Figure 4.7 Notations of the CDPR

Table 4.7 Parameters of the CDPR

Parameter	**Value**
Mass of the end-effector	3kg
Moment of inertia of end-effector about *x* axis of F_e	$0.03\text{kg} \cdot \text{m}^2$
Moment of inertia of end-effector about *y* axis of F_e	$0.03\ \text{kg} \cdot \text{m}^2$
Moment of inertia of end-effector about *z* axis of F_e	$0.005\ \text{kg} \cdot \text{m}^2$

The model of the CDPR is created in the Gazebo simulator as shown in Figure 3.9 on page 58. A cable tension is applied between a pair of anchor points of the CDPR. The control simulation is conducted in the Gazebo simulator using a computer with an Intel i7-7700HQ CPU and an 8-Gigabyte memory. Graphics Processing Unit is not used to train the FNN. The frequency of the simulation is 100 Hz.

To have data samples more uniformly distributed (i.e., data samples are more representative to wrenches and poses when controlling the CDPR), two requirements are set when recording data samples: 1) the norm of the pose vector $\boldsymbol{x} = \boldsymbol{\Omega} = [\phi, \theta, \psi]^T$ changes from the last data sample by 2.86 degrees or more; or 2) the norm of the wrench vector $\boldsymbol{w}$ changes from the last data sample by 5 Nm or more. In this way, the recorded data samples are sparser and more uniformly distributed and thus more representative to wrenches and poses when controlling the CDPR.

Three FNNs (i.e., FNNs I-III) with one hidden layer but different numbers of hidden layer units are investigated in the simulation. Each FNN is trained and validated using a certain number of data samples. Data samples used for training an FNN are not used for validating the FNN. The number of data samples used for training quadruples that for validation. In the simulation, all three FNNs are trained by the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.1 and a maximum number of iterations of 51,200. The training time and the mean square errors of the three FNNs are listed in Table 4.8. The more data samples or the more hidden layer units, the longer time to train the FNNs.

Table 4.8 Parameters of FNNs

Parameter	**FNN I**	**FNN II**	**FNN III**
Number of data samples used in training	256	1024	1024
Number of hidden layer units	32	32	128
Training time (unit: sec)	163.1	227.1	300.9
Mean square error in training	0.0961	0.2935	0.0564
Mean square error in validation	1.258	0.4743	0.5855

After the three FNNs are trained, each FNN is paired up with the same PID controller to form a PID-FNN control strategy to control the CDPR to track a target trajectory. For performance comparison, the same PID controller is also paired up with the known Jacobian of the CDPR to control the CDPR to track a target trajectory. The control gains of the PID controller are $k_P = 10$, $k_I = 0.5$, and $k_D = 0.1$.

4.2.2 Simulation Results

The pitch, roll, and yaw of the CDPR, tracking errors, cable tensions, and the norm of disturbance wrench $\boldsymbol{w}_N$ caused by the approximation error of every FNN are plotted in Figures 4.8-4.19. As shown in Figures 4.8-4.10, the CDPR can be controlled to track the target trajectory using the PID–FNN control strategy with FNNs I-III and the same PID controller plus the known Jacobian. If there is no disturbance, the PID controller plus the known Jacobian can control the CDPR to track the target trajectory with negligible errors. However, according to the tracking errors shown in Figures 4.11-4.13, when subject to a random disturbance generating a disturbance wrench ranging from 0 to 0.1 Nm, the PID–FNN control strategy with FNNs I-III outperforms the PID controller plus the known Jacobian in tracking the target trajectory. A possible reason for this phenomenon is that the data samples used to train the FNNs have already taken into account the random disturbance, leading the FNNs to be adaptive to the random disturbance. The adaptability of a learning-based control strategy to the environment was also observed in (Anusha Nagabandi et al., 2017).

Cable tensions of the CDPR controlled by the PID–FNN control strategy with FNNs I-III and the same PID controller plus the known Jacobian are shown in Figures 4.14-4.18. As shown in Figure 4.17 and Figure 4.18, at any time instant, at least one cable tension is zero. This is because, with the PID controller plus the known Jacobian, the optimal cable tension distribution (Otis et al., 2009) can be achieved by solving the inverse dynamics equation. However, as shown in Figures 4.14-4.16, the cable tensions are all non-zero at some time instants. This suggests that cable tensions derived from the FNNs do not always achieve the optimal cable tension distribution. It makes sense because cable tensions of data samples used to train the FNNs do not necessary obey the optimal cable tension distribution. Thus, cable tensions derived from the FNNs do not necessarily obey the optimal cable tension distribution.

As shown in Figures 4.8-4.10, although the trajectories of the CDPR controlled by the PID-FNN control strategy with different FNNs are close to the given trajectory, there are errors. Because the trajectories of the CDPR controlled by different control strategies are different, Jacobians of the CDPR controlled by different control strategies are different in every moment. As a result, even if the target wrench of the CDPR is the same, the corresponding sets of target cable tensions of the

CDPR are generally different. Therefore, to study the approximation errors of the FNNs in the simulation, one cannot simply compare the sets of cable tensions of the CDPR controlled by the PID–FNN control strategy with FNNs I-III to that of the CDPR controlled by the PID controller plus the known Jacobian. Alternatively, to study the approximation error of an FNN in the simulation, one can analyze the disturbance wrench (i.e., $\boldsymbol{w}_N$) caused by the approximation error of an FNN. $\boldsymbol{w}_N$ is the difference between the wrench generated by cable tensions derived from an FNN and the wrench generated by cable tensions calculated by solving the inverse dynamics equation. The norms of $\boldsymbol{w}_N$ for FNNs I-III are shown in Figure 4.19. It can be seen that the norms of $\boldsymbol{w}_N$ are bounded indeed. According to the discussion in the last paragraph of section 3.2.1.3, the PID–FNN control strategy with FNNs I-III is stable of the form of UUB in this case.

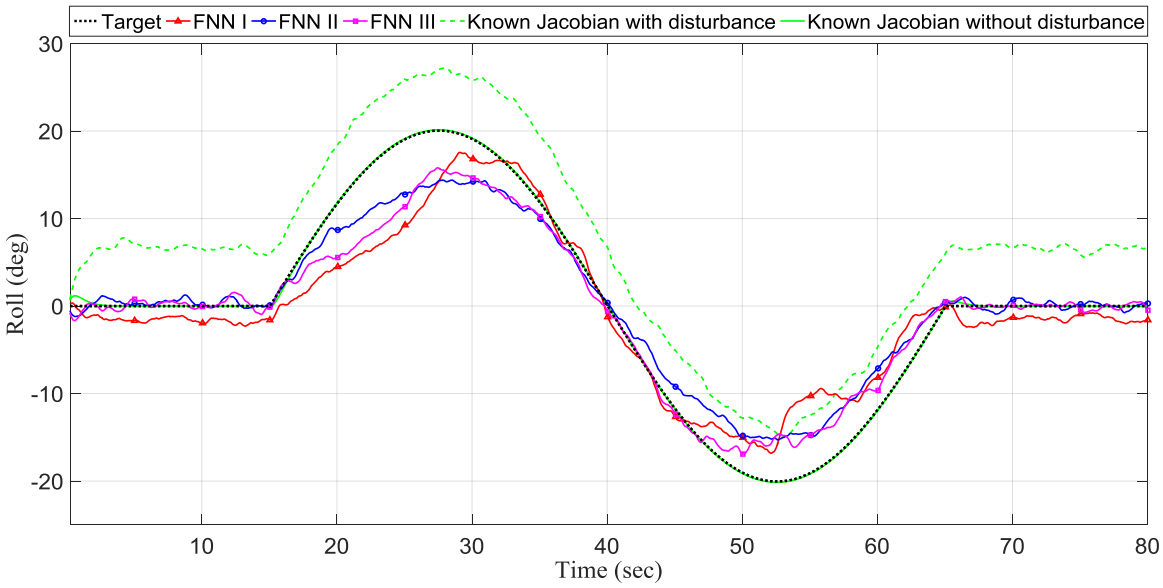


Figure 4.8 Tracking the roll of the CDPR

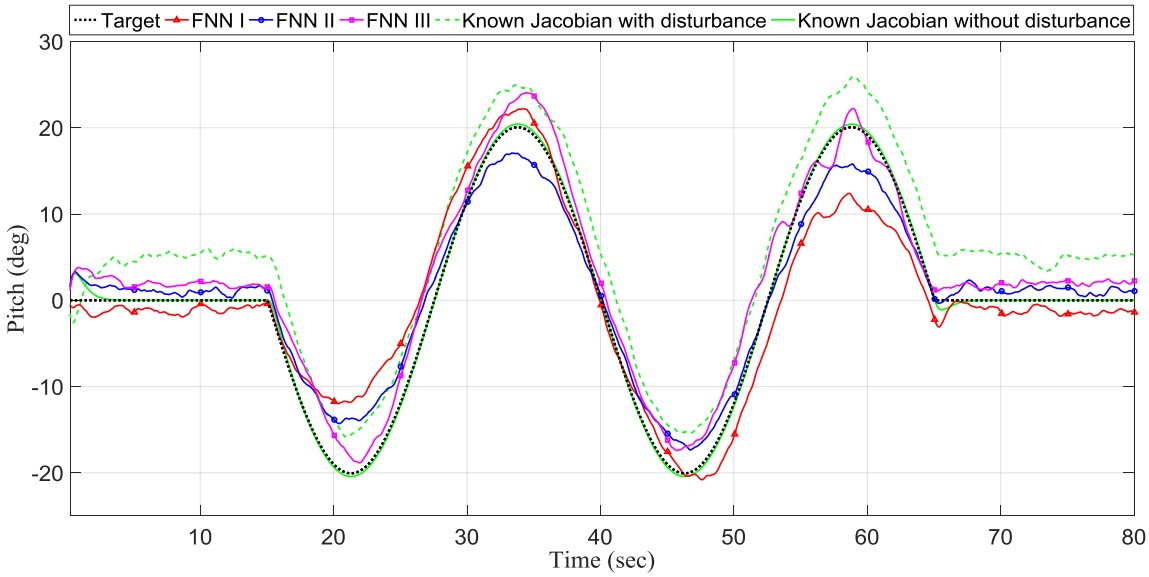


Figure 4.9 Tracking the pitch of the CDPR

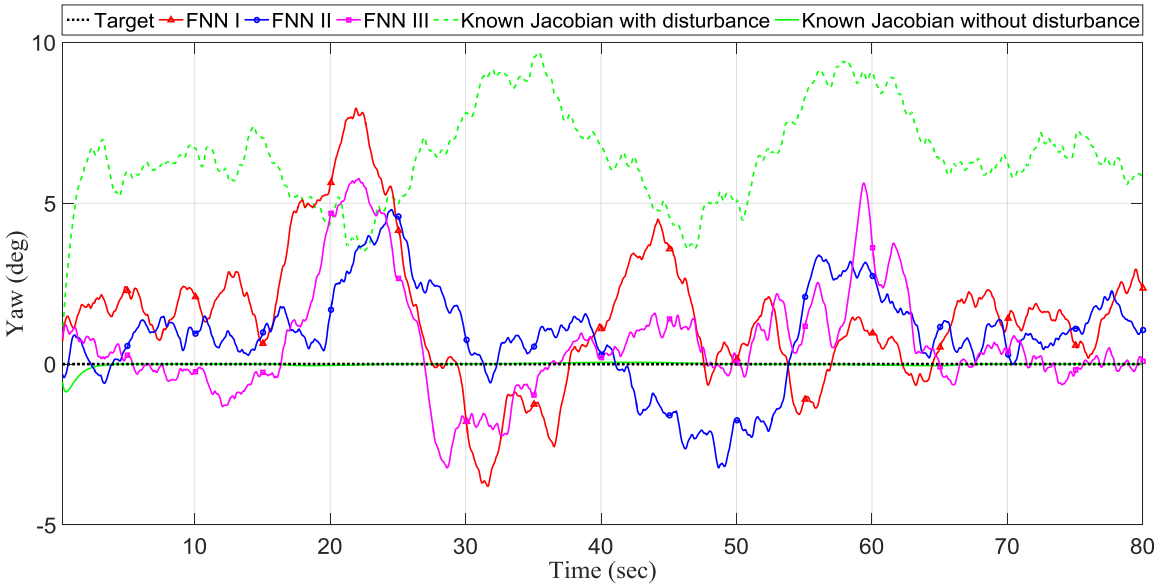


Figure 4.10 Tracking the yaw of the CDPR

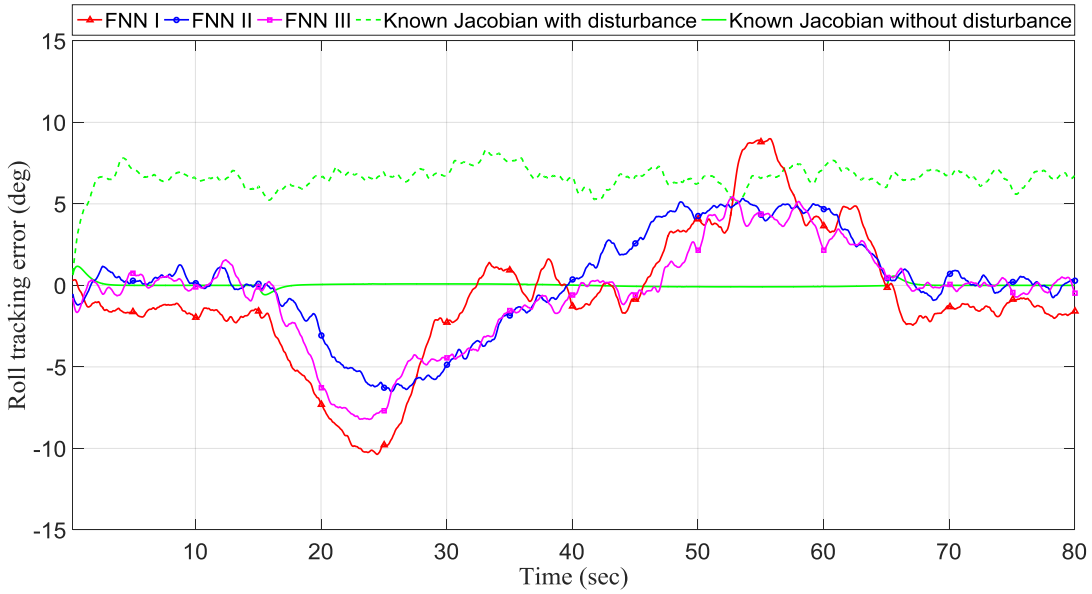


Figure 4.11 Roll tracking errors

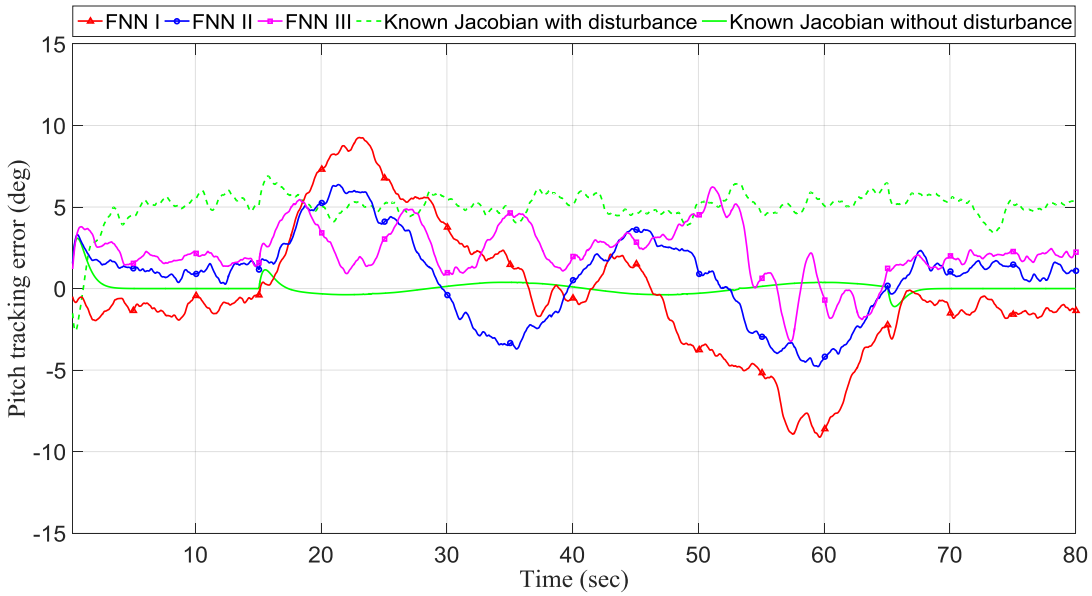


Figure 4.12 Pitch tracking errors

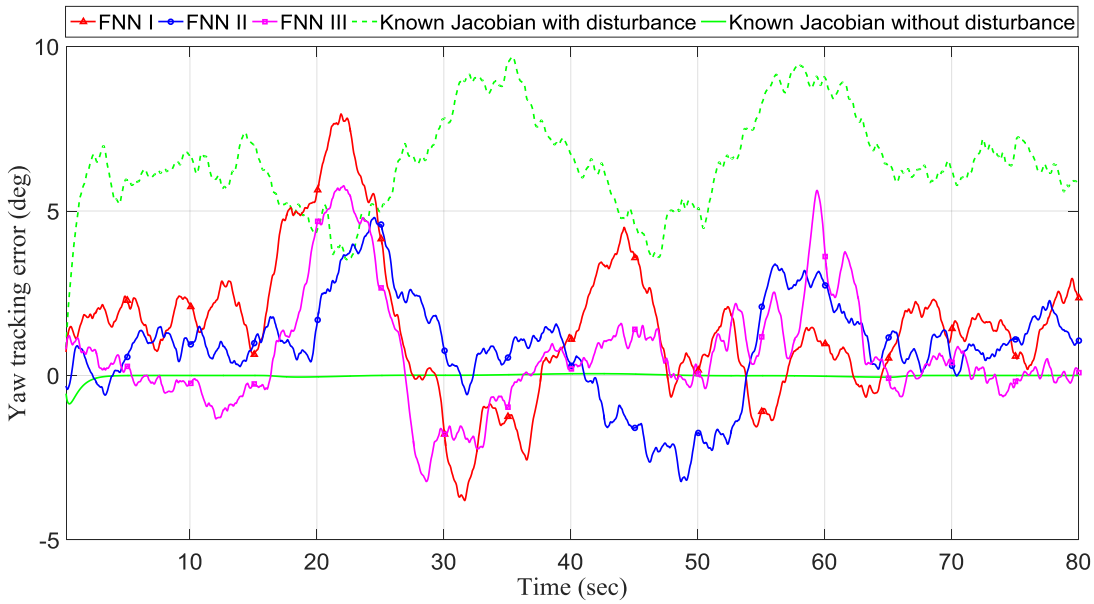


Figure 4.13 Yaw tracking errors

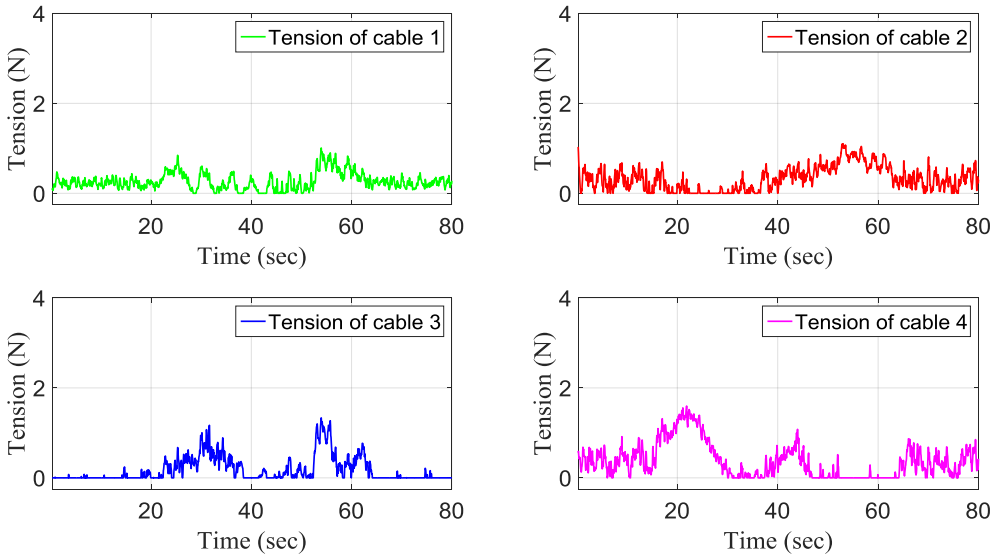


Figure 4.14 Cable tensions of the CDPR with FNN I

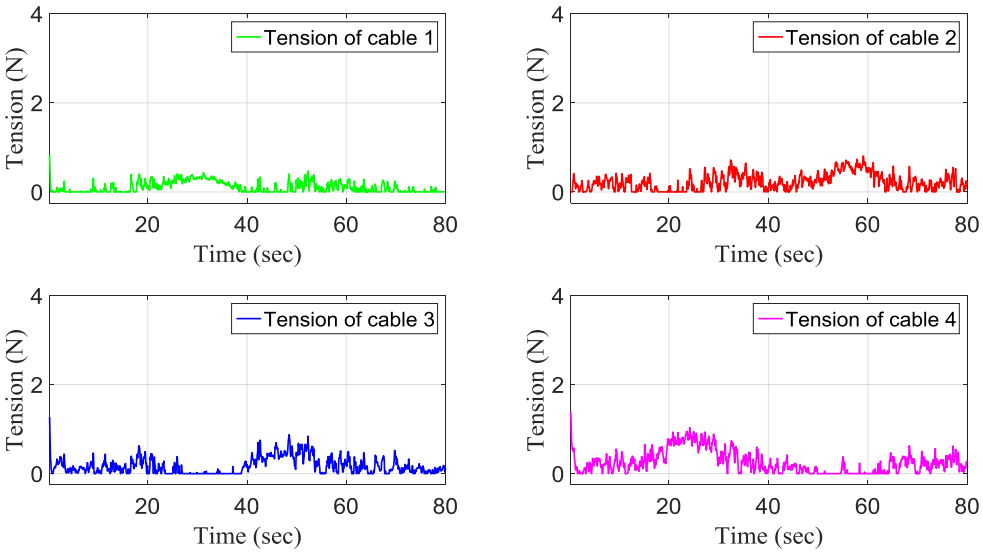


Figure 4.15 Cable tensions of the CDPR with FNN II

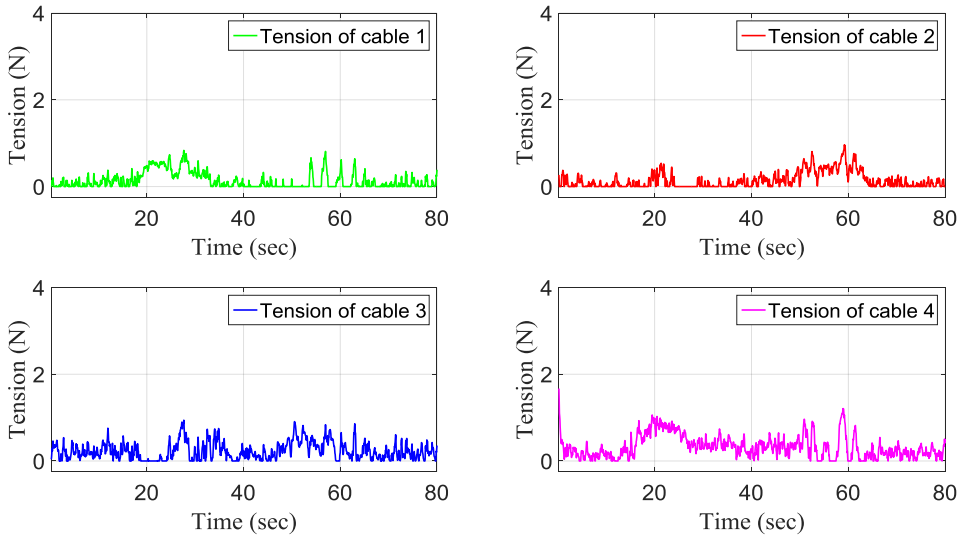


Figure 4.16 Cable tensions of the CDPR with FNN III

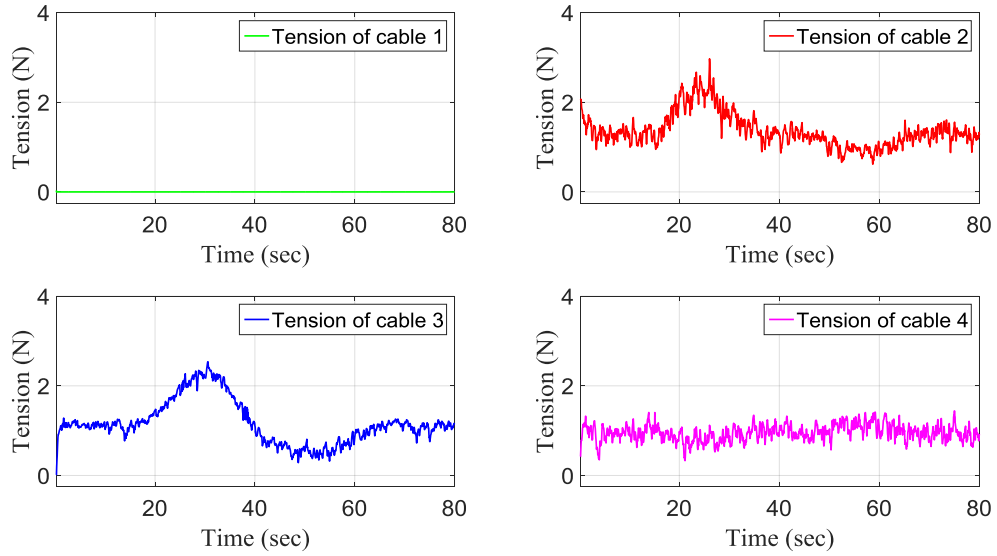


Figure 4.17 Cable tensions of the CDPR with known Jacobian and disturbance

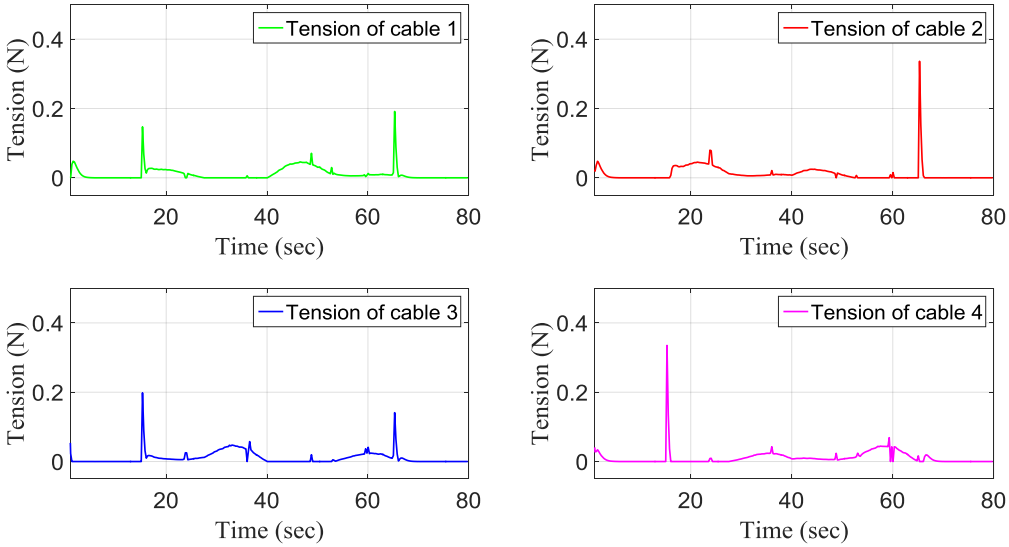


Figure 4.18 Cable tensions of the CDPR with known Jacobian but without disturbance

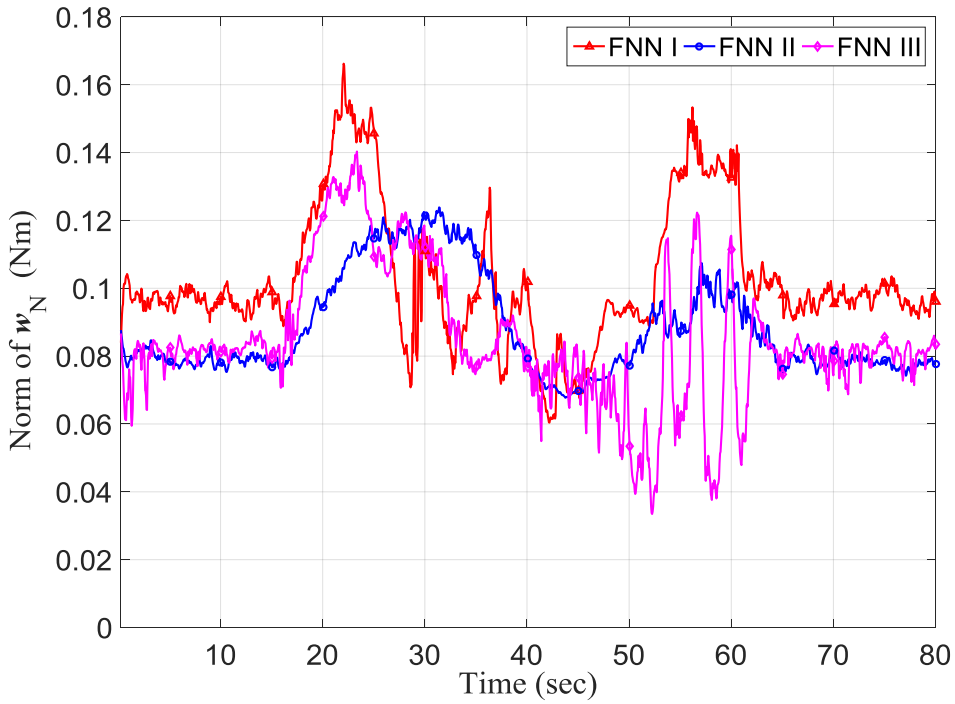


Figure 4.19 Norm of approximation error of FNNs

4.2.3 Conclusion

Simulation results show that the FNN can help the PID–FNN control strategy control the CDPR to track a target trajectory. However, the PID–FNN control strategy also has limitations. One of the limitations is that the ANN of this control framework needs to be trained before it can be used together with the robust controller to control a CDPR. Moreover, the requirements of the number of hidden layer units of the ANN and the number of data samples used to train the ANN can vary from case to case. Thus, one may try a few times to obtain the proper number of hidden layer units of the ANN and the proper number of data samples used to train the ANN for a specific application.

4.3 Learning Control Strategies Based on Deep Reinforcement Learning

For the study, simulations are conducted to evaluate DDPG-based strategies in control a CDPR. Polices included in strategies are trained by DDPG at first. Then, cable tensions of the CDPR in reaching a randomly selected target pose are studied. The robustness of DDPG-based strategies to model uncertainties is evaluated in a case study as well.

4.3.1 Training of DDPG in Control Strategies

The hybrid DDPG strategy and the end-to-end DDPG strategy have to be trained before they can be used to control a CDPR. Training a DRL algorithm on a real robot may damage the robot, especially when the DRL strategy has not been well-trained in the initial stage. Thus, a robot model in a simulator, rather than a real robot, is highly desirable as simulations are always safe for the real robot (Kober et al., 2013). Moreover, a simulator is also helpful in developing and fine-tuning the DRL strategy because the iteration of simulations is much faster than that of real experiments (Kober et al., 2013). DDPG in the proposed strategies is trained to control an example CDPR using the Gazebo simulator in this section.

As shown in Figure 4.20 on page 80, the example CDPR has three rotational DOFs and four cables. F_b represents the base frame and F_e represents the end-effector frame. The pose of the end-effector with respect to the base (i.e., the pose of the CDPR) is described by a vector of three Euler angles $[\phi\ \theta\ \psi]$ with a ψ-θ-ϕ (i.e., yaw-pitch-roll) sequence (Craig, 2009). DDPG in the proposed strategies is trained within a workspace, called training workspace, of the CDPR. The training workspace in this study is the set of poses that the CDPR can reach with the roll, pitch, and yaw of the CDPR within $[-20, 20]$ (unit: deg). Gravity is neglected in this study to simplify the analysis.

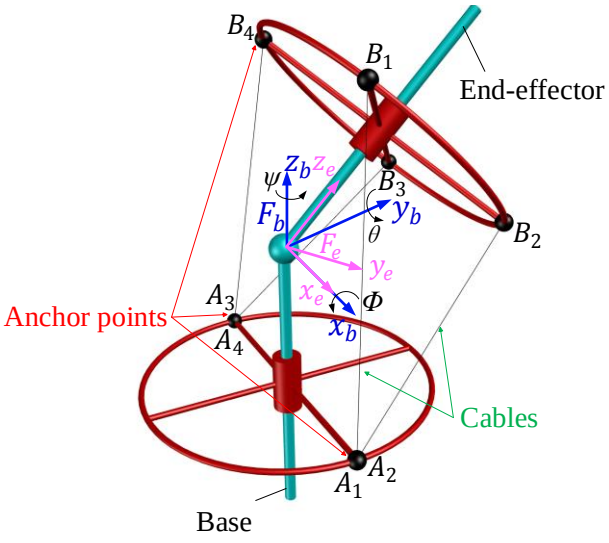


Figure 4.20 Notations of the example CDPR

A model of the example CDPR is created in the Gazebo simulator, as shown in Figure 3.9. The four cables of the CDPR are not shown in Figure 3.9. Cable tension is applied between a pair of anchor points of the CDPR. The parameters and the positions of the anchor points of the CDPR are shown in Table 4.9 and Table 4.10, respectively.

Table 4.9 Parameters of the CDPR

Parameter	**Value**
Mass of the end-effector	3 kg
Moment of inertia of end-effector about *x* axis of F_e	$0.02\ \text{kg} \cdot \text{m}^2$
Moment of inertia of end-effector about *y* axis of F_e	$0.02\ \text{kg} \cdot \text{m}^2$
Moment of inertia of end-effector about *z* axis of F_e	$0.02\ \text{kg} \cdot \text{m}^2$

Table 4.10 Positions of anchor points of the CDPR (unit: m)

Pair of anchor points	**Positions of anchor points on the base**	**Positions of anchor points on the end-effector**
1	A_1: $[0.1, 0.0, -0.15]^T$	B_1: $[0.0707, -0.0707, 0.05]^T$
2	A_2: $[0.1, 0.0, -0.15]^T$	B_2: $[0.0707, 0.0707, 0.05]^T$
3	A_3: $[-0.1, 0.0, -0.15]^T$	B_3: $[-0.0707, 0.0707, 0.05]^T$
4	A_4: $[-0.1, 0.0, -0.15]^T$	B_4: $[-0.0707, -0.0707, 0.05]^T$

4.3.1.1 Training Setup

For the study, the state variables of the hybrid DDPG strategy are the same as those of the end-to-end DDPG strategy. The state variables of the proposed strategies at time step *t* can be expressed as

$$\boldsymbol{s}_t = [\boldsymbol{X} \quad \dot{\boldsymbol{X}} \quad \Delta\boldsymbol{X}] \tag{4.1}$$

where $\boldsymbol{X} = [\phi \quad \theta \quad \psi]$ is the pose of the CDPR at time step *t*. $\dot{\boldsymbol{X}} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]$ is the angular velocity of the CDPR at time step *t*. $\Delta\boldsymbol{X}$ is the difference between the target pose of the CDPR, denoted as $\boldsymbol{X}^*$, and the pose of the CDPR at time step *t*. Thus, $\Delta\boldsymbol{X}$ can be expressed as

$$\Delta\boldsymbol{X} = \boldsymbol{X}^* - \boldsymbol{X} \tag{4.2}$$

It should be noted that $\boldsymbol{s}_t$ is a vector having nine scalar elements in this study.

The action variables of the hybrid DDPG strategy are different from those of the end-to-end DDPG strategy. The action variables of the hybrid DDPG strategy at time step *t* can be expressed as

$$\boldsymbol{a}_t^w = [w_x \quad w_y \quad w_z] \tag{4.3}$$

where w_x, w_y, and w_z represent the target torques about the *x*, *y*, and *z* axes of F_e, respectively. The action variables of the end-to-end DDPG strategy at time step *t* can be expressed as

$$\boldsymbol{a}_t^\tau = [\tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4] \tag{4.4}$$

where τ_i $(i = 1, 2, 3, 4)$ is the target cable tension in the *i*th cable.

Rewards of the hybrid DDPG strategy and the end-to-end DDPG strategy used in this study are defined below. For the hybrid DDPG strategy, the reward is designed as

$$r_t^w = -2\|\Delta \boldsymbol{X}\| - \left\|\dot{\boldsymbol{X}}\right\| \tag{4.5}$$

where $\|*\|$ represents the Euclidean norm of $*$. r_t^w has two terms. The first term is defined by the difference between the target pose and the pose of the CDPR at time step *t*. The second term is defined by the velocity of the CDPR at time step *t*. In this way, a larger reward is granted if the CDPR is closer to the target pose or the velocity of the CDPR is smaller. If the CDPR reaches the target pose with a full stop, the maximum reward (i.e., $r_t^w = 0$) is granted.

For the end-to-end DDPG strategy, whether the strategy can learn the optimal tension distribution of cables or not depends on the reward. Thus, two rewards are designed and tested, aiming to find a reward with which the end-to-end DDPG strategy can learn the optimal tension distribution of cables. The two rewards of the end-to-end DDPG strategy are designed as

$$r_t^{\tau 1} = -2\|\Delta \boldsymbol{X}\| - \left\|\dot{\boldsymbol{X}}\right\| \tag{4.6}$$

$$r_t^{\tau 2} = -2\|\Delta \boldsymbol{X}\| - \left\|\dot{\boldsymbol{X}}\right\| - 0.2\|\boldsymbol{a}_t^\tau\| \tag{4.7}$$

where $r_t^{\tau 1}$ is the same as r_t^w. Compared to $r_t^{\tau 1}$, $r_t^{\tau 2}$ has a third term, namely, the Euclidean norm of the vector of cable tensions or the action variables of the end-to-end DDPG strategy. Such a cable tension term gives higher rewards to actions producing smaller tensions. The design of $r_t^{\tau 2}$ is inspired by the reward consisting of both balancing and goal-oriented terms for a simulated bicycle riding task in (Randløv & Alstrøm, 1998). With $r_t^{\tau 2}$, a larger reward is granted if the CDPR is closer to the target pose; the velocity of the CDPR is smaller; or the Euclidean norm of the vector of cable tensions is smaller. In this manner, the end-to-end-DDPG strategy's goal is not only to

accomplish the task of controlling a CDPR, but also achieve the optimal tension distribution of cables. r_t^w, $r_t^{\tau 1}$, and $r_t^{\tau 2}$ are possible to be further optimized according to studies of multi-objective reinforcement learning that aims to find compromising solutions balancing different objectives to RL problems (Moffaert, Drugan, & Nowé, 2013). Multi-objective reinforcement learning has been successfully applied to control variable speed wind turbines achieving the optimal balance of power generation stability and rotor angular speed in (Fernandez-Gauna, Fernandez-Gamiz, & Grana, 2017).

For the study, the four networks of the DDPG used by both the hybrid DDPG strategy and the end-to-end DDPG strategy are fully connected neural networks with two hidden layers. The architecture of the four DNNs is shown in Table 4.11 and the hyper-parameters used to train the DDPG are shown in Table 4.12. The architecture of the four DNNs of the DDPG and the hyper-parameters used by both the hybrid DDPG strategy and the end-to-end DDPG strategy are the same except the number of outputs of the actor network and the actor-target network and the number of inputs of the critic network and the critic-target network. N_a represents the number of actions. For the study, $N_a = 3$ for the hybrid DDPG strategy and $N_a = 4$ for the end-to-end DDPG strategy. The learning rates of the critic network and the critic-target network change from 0.0001 to 0.00005 if the training loss is less than 0.001. The data (i.e., states, actions, reward, and next states) used to train DDPG in both the hybrid DDPG strategy and the end-to-end DDPG strategy are collected from the Gazebo simulator. For each episode, the CDPR starts at $\boldsymbol{X} = [0 \quad 0 \quad 0]$ (unit: deg). An episode ends if the CDPR reaches a target pose, the maximum number of 1,000 time steps is reached, or the CDPR goes out of the training workspace.

Table 4.11 Architecture of the four networks of the DDPG

	Actor and actor-target networks	**Critic and critic-target networks**
Number of inputs	9	$9 + N_a$
Activation function 1	ReLU	ReLU
Number of units in layer 1	90	150
Activation function 2	ReLU	ReLU
Number of units in layer 2	60	120
Activation function 3	tanh	None
Number of outputs	N_a	1

Table 4.12 Hyper-parameters used to train the DDPG

Hyper-Parameter	**Value**
Maximum number of time steps for each episode	1000
Learning rate of the actor and actor-target networks	0.001
Learning rate of the critic and critic-target networks	$0.0001 \rightarrow 0.00005$
Discount factor γ	0.95
Update rate (target) τ	0.001
Size of the experience replay buffer	10000
Size of mini-batch	1024
Maximum magnitude of random exploration noise (i.e., an element of $\boldsymbol{N}_t$)	$\frac{\text{range of an action}}{\text{number of episodes} + 2}$

4.3.1.2 Training

For the study, it is assumed that elements of $\boldsymbol{a}_t^{\tau}$ (i.e., the individual cable tensions) are within $[0,1]$ (unit: N) and elements of $\boldsymbol{a}_t^{w}$ (i.e., the target wrenches about the *x*, *y*, and *z* axes of F_e) are within $[-0.1, 0.1]$ (unit: Nm). The Adam optimizer (Kingma & Ba, 2014) is utilized in training. With the training setup shown in Table 4.12, DDPG in the proposed strategies is trained based on a CDPR model in the Gazebo simulator. The frequency of the Gazebo simulator is set to 100 Hz and the time step of the DDPG is 0.1 second. The average rewards in every episode are shown in Figure 4.21. It is shown that the hybrid DDPG strategy converges within 70 episodes, while the end-to-

end DDPG strategy with $r_t^{\tau 1}$ and $r_t^{\tau 2}$ converges within 100 and 400 episodes, respectively. Therefore, the hybrid DDPG strategy learns faster than the end-to-end DDPG strategy even when they use the same reward.

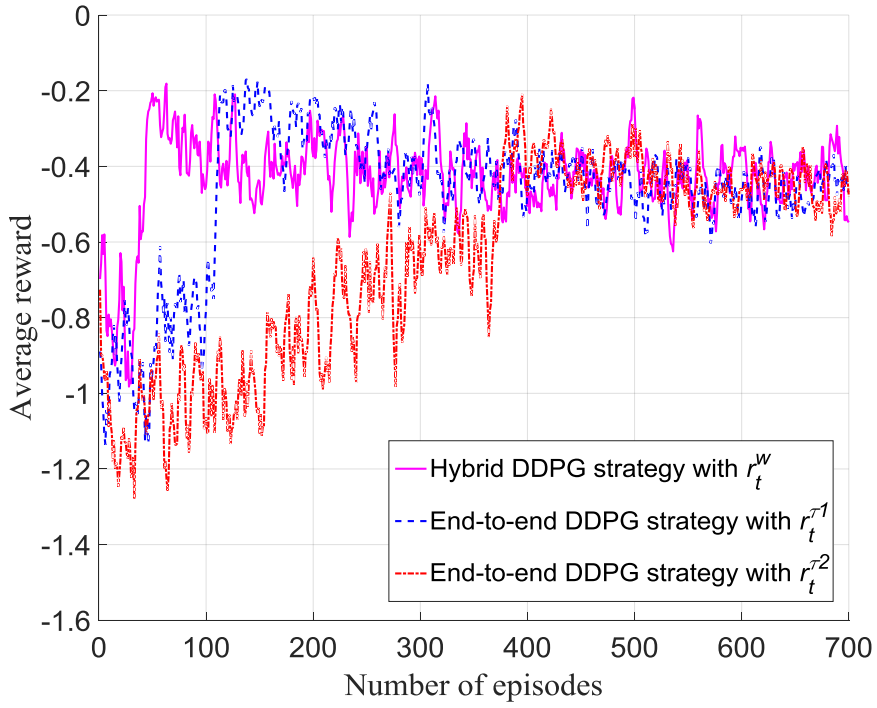


Figure 4.21 Average rewards of DDPG in the proposed strategies

According to (Kober et al., 2013), a complicated task becomes easier to learn if some of its subtasks have already been accomplished. For the hybrid DDPG strategy, the second subtask of converting the target wrench in the task space to a set of target cable tensions in the joint space is accomplished by solving the inverse dynamics equation. Thus, the DDPG of the hybrid DDPG strategy can focus on the first subtask while the DDPG of the end-to-end DDPG strategy has to deal with both the first and the second subtasks. This explains why the hybrid DDPG strategy learns faster than the end-to-end DDPG strategy. Moreover, the number of actions of the DDPG in the end-to-end DDPG strategy is four while that in the hybrid DDPG strategy is three. The reduced number of actions is another possible reason that the hybrid DDPG strategy learns faster than the end-to-end DDPG strategy.

4.3.2 Optimal Tension Distribution of Cables

The example CDPR concerned in this study is a fully-constrained CDPR. A fully-constrained CDPR in its force-closure workspace has an infinite number of feasible tension distributions. Therefore, how to obtain the optimal tension distribution (Fang, Franitza, Torlo, Bekes, & Hiller, 2004; Oh & Agrawal, 2005; Otis et al., 2009) of cables is studied for the example CDPR in this section. The hybrid DDPG strategy calculates the optimal tension distribution of cables by solving the inverse dynamics equation. The end-to-end DDPG strategy has to learn the optimal tension distribution of cables on its own. This section demonstrates that, with a proper reward, the end-to-end DDPRG strategy can learn the optimal tension distribution of cables as well as the hybrid DDPG strategy obtains the optimal tension distribution of cables using a non-learning-based approach.

The tension distributions of the CDPR controlled by the end-to-end DDPG strategy with $r_t^{\tau 1}$ and $r_t^{\tau 2}$ in reaching a randomly selected target pose $\boldsymbol{X} = [9 \;\; 12 \;\; 15]$ (unit: deg) is studied. The end-to-end DDPG strategy with $r_t^{\tau 2}$ whose third term is about cable tensions is expected to be able to learn the optimal tension distribution of cables. The tension distributions of the end-to-end DDPG strategy with $r_t^{\tau 1}$ and $r_t^{\tau 2}$ are shown in Figure 4.22 and Figure 4.23, respectively. The optimal tension distribution (i.e., the dash lines in Figure 4.22 and Figure 4.23) is obtained by solving the inverse dynamics equation. It is shown that the end-to-end DDPG strategy with $r_t^{\tau 2}$ is able to learn the optimal tension distribution, while the end-to-end DDPG strategy with $r_t^{\tau 1}$ cannot. Therefore, with a proper reward (e.g., $r_t^{\tau 2}$), the end-to-end DDPG strategy can learn the optimal tension distribution of a fully-constrained CDPR.

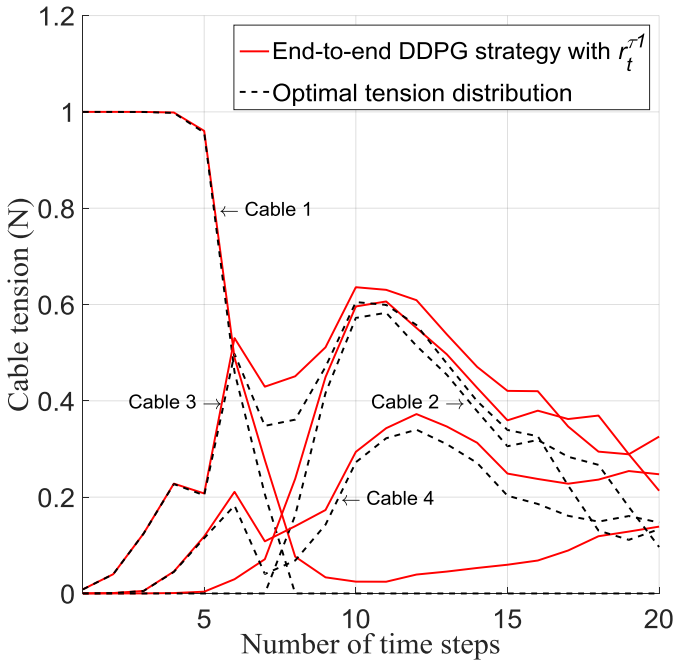


Figure 4.22 Cable tensions of the CDPR

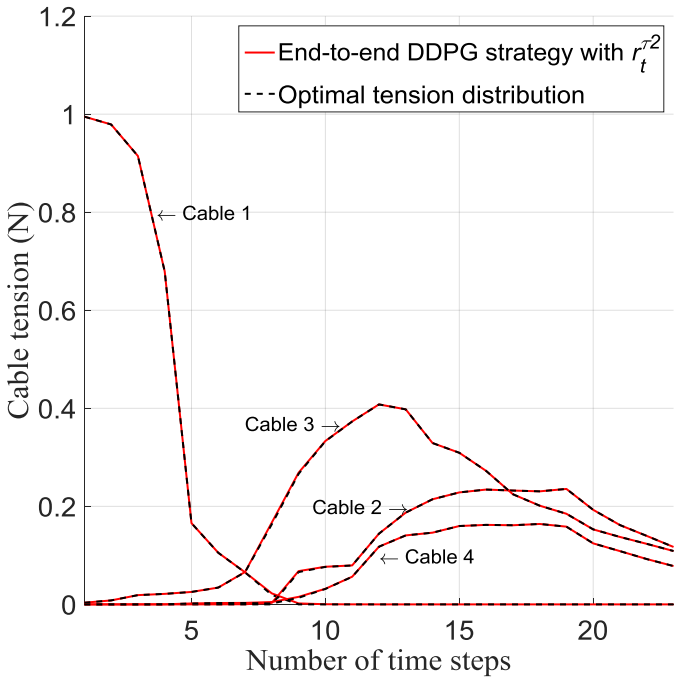


Figure 4.23 Cable tensions of the CDPR

4.3.3 Robustness to Model Uncertainty

The model of a robot used to train a DRL algorithm may not capture all the details of the real robot in practice (Kober et al., 2013). A setup or model of the CDPR to be controlled by the proposed strategies may be slightly different from the setup or model of the same CDPR based on which DDPG in the proposed strategies is trained. It would be ideal if the proposed strategies whose DDPG is trained using one model of the CDPR can be used to control the CDPR whose model has slightly changed. Such an adaptive capability requires the proposed strategies to be robust to model uncertainties (Kober et al., 2013).

The robustness of the proposed strategies to model uncertainties is investigated in this section based on a pose-tracking test and a trajectory-tracking test. The proposed strategies whose DDPG is trained using one of the models of the CDPR in the Gazebo simulator are first used to control the CDPR with the same model. In this case, there is no model uncertainty or difference between the model based on which DDPG is trained and the model to be controlled by DDPG. This works as the baseline for the study of the robustness of the proposed strategies to model uncertainties. Then, the proposed strategies are used to control the CDPR whose model has been slightly changed to test the robustness of the proposed strategies to model uncertainties.

For the pose-tracking test, the CDPR is controlled by the proposed strategies to move from $\boldsymbol{X} = [0 \;\; 0 \;\; 0]$ (unit: deg) to 1,000 randomly selected target poses within the training workspace, as shown in Figure 4.24. A pose-tracking test is considered successful if the CDPR is able to reach a target pose (i.e., the Euclidean norm of the tracking error defined by $\|\Delta \boldsymbol{X}\|$ is less than 1.8 degrees) and it takes no more than 150 time steps (i.e., 15 seconds). Otherwise, the pose-tracking test is considered failed.

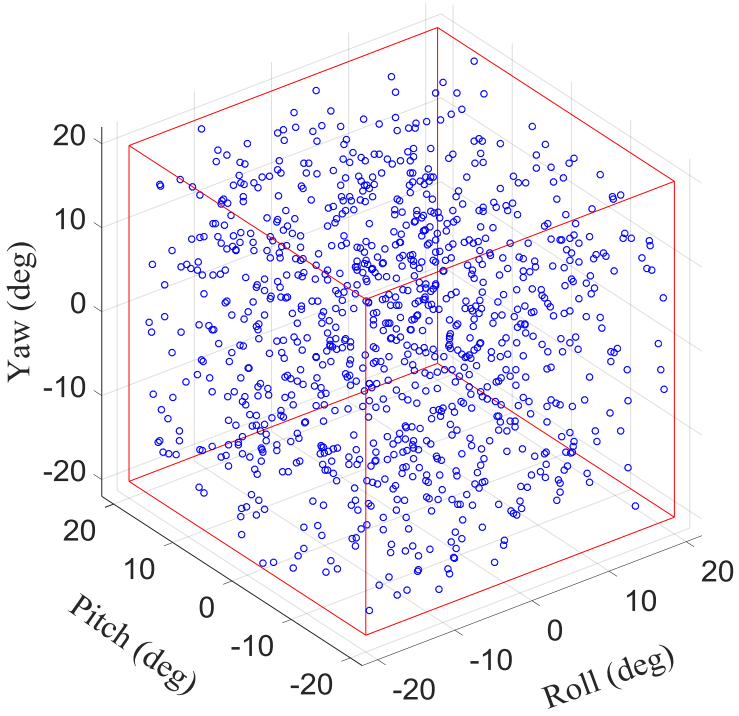


Figure 4.24 Randomly selected target poses

In the trajectory-tracking test, the target trajectory to be tracked is designed as

$$\begin{cases} \phi = 45\sin(0.01\pi j)/\pi \\ \theta = 22.5\cos(0.01\pi j)/\pi - 22.5/\pi \\ \psi = 45\sin(0.02\pi j)/\pi \end{cases} \tag{4.8}$$

where $j = 0,1,\ldots,599$ and j is updated in every 15 time steps (i.e., 1.5 seconds).

4.3.3.1 Evaluate Control Strategies on a CDPR without Model Uncertainty

The proposed strategies whose DDPG is trained using a model of the CDPR in the Gazebo simulator are used to control the CDPR with the same model. For the proposed strategies, the model of the CDPR based on which DDPG in the strategies is trained is regarded as the model of the CDPR without model uncertainty. The outcomes of the pose-tracking test are shown in Table 4.13. The success rate of the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$ is slightly lower than those of the hybrid DDPG strategy and the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$. To investigate why the proposed strategies cannot control the CDPR to reach a target pose, the target poses that the proposed strategies cannot control the CDPR to reach are plotted in Figure 4.25. It turns out that all target poses the CDPR fails to reach are close to the boundary of the

training workspace. If a target pose is close to the boundary of the training workspace, the CDPR may move out of the training workspace when moving towards such a target pose. If the CDPR is out of the training workspace, it is very likely that the CDPR cannot move back to the training workspace because the training of DDPG in the strategies is limited to the training workspace. Thus, poses close to the boundary of the training workspace are more challenging to track than those close to the center of the training workspace. This suggests that the training workspace should be larger than the workspace used in a control task in practice.

Table 4.13 Outcomes of the pose-tracking test

Strategy	**Number of target poses the CDPR reaches successfully**	**Number of target poses the CDPR fails to reach**	**Success rate**
Hybrid DDPG strategy with the reward of r_t^w	993	7	99.3%
End-to-end DDPG strategy with the reward of $r_t^{\tau 1}$	978	22	97.8%
End-to-end DDPG strategy with the reward of $r_t^{\tau 2}$	995	5	99.5%

Figure 4.26 shows the trajectories of the CDPR controlled by the proposed strategies to track the target trajectory defined in (4.8). The proposed strategies can control the CDPR to track the target trajectory indeed. Figure 4.27 shows the Euclidean norm of the tracking error when the CDPR is controlled to track the target trajectory. The hybrid DDPG strategy and the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$ have smaller tracking error than the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$. Since the magnitude of cable tensions is included in the reward of $r_t^{\tau 2}$, the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$ tends to achieve not only smaller tracking error and velocity, but also smaller cable tensions. It makes sense that the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$ leads to a larger tracking error than the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$ and the hybrid DDPG strategy with the reward of r_t^w.

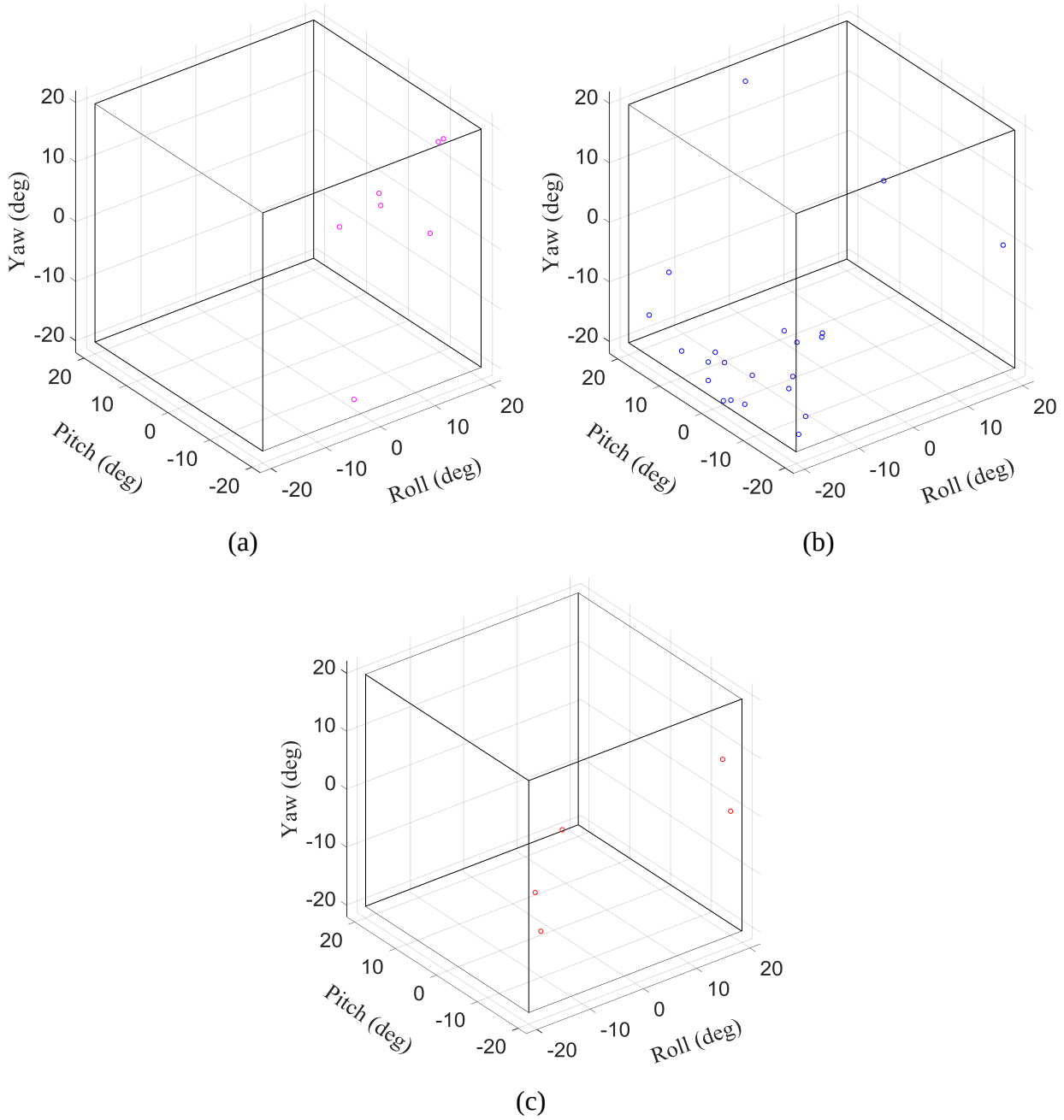


Figure 4.25 Target poses fails to be reached by the CDPR without model uncertainty using: (a) the hybrid DDPG strategy; (b) the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$; and (c) the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$

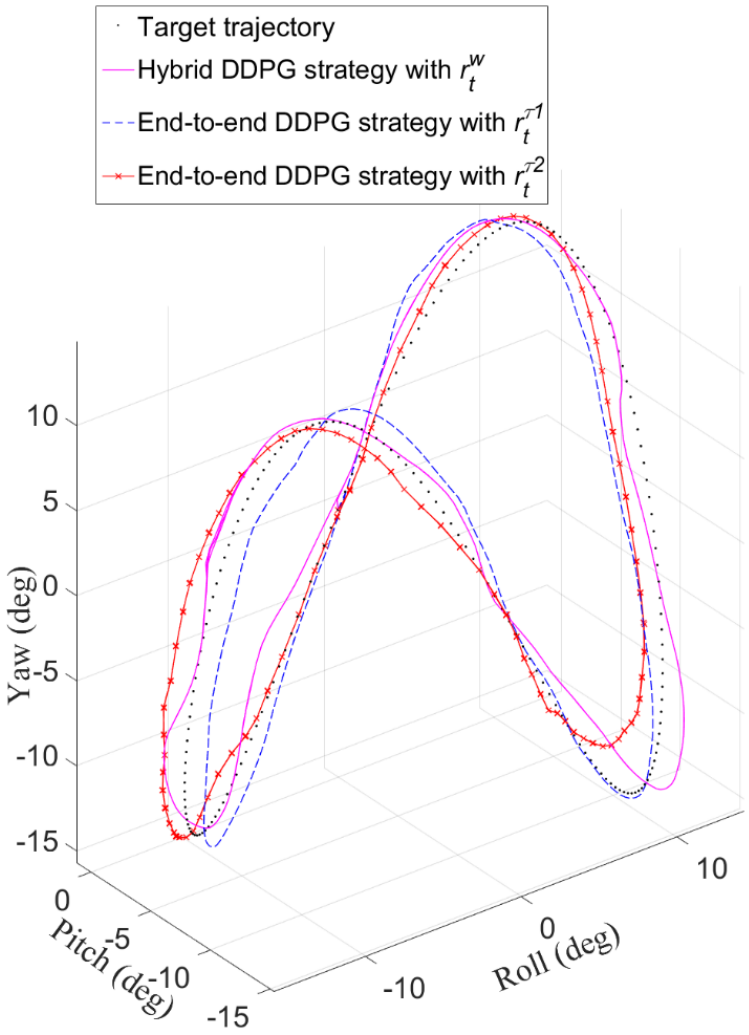


Figure 4.26 Trajectories of the CDPR without model uncertainty

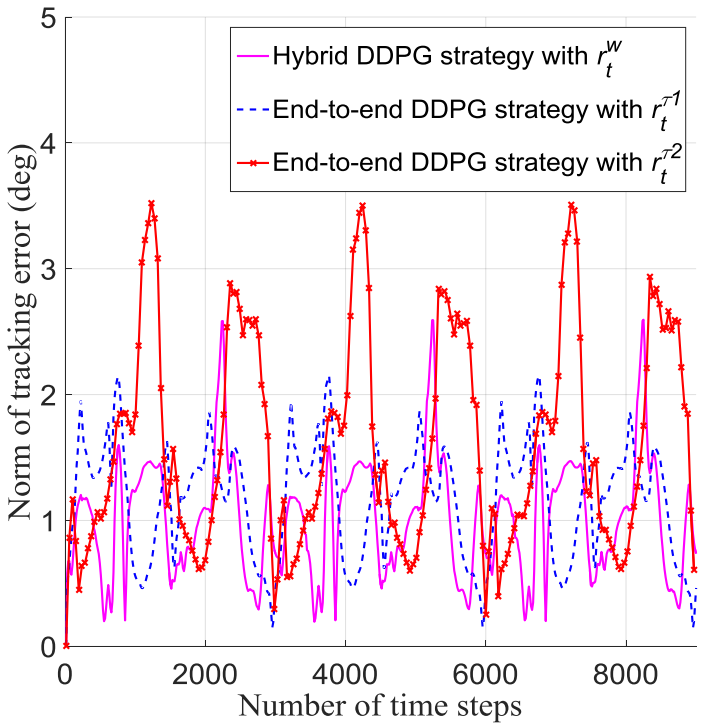


Figure 4.27 Tracking error of the CDPR without model uncertainty

4.3.3.2 Evaluate Control Strategies on a CDPR with Model Uncertainties

The proposed strategies are used to control the CDPR with model uncertainties in this section. In other words, the model of the CDPR to be controlled is slightly different from the model of the CDPR based on which DDPG in the strategies is trained.

For the study, the model uncertainties are supposed to be caused by position errors of anchor points of the CDPR, as shown in Figure 4.28. The positions of anchor points of the CDPR without model uncertainty are shown in Table 4.10, while the positions of anchor points of the CDPR with model uncertainties are shown in Table 4.14. The anchor points on the end-effector of the CDPR with model uncertainties are supposed to rotate 5.73 degrees about the z axis of F_e and slide 0.01 meter along the z axis of F_e, compared to those of the CDPR without model uncertainty. Moreover, the moment of inertias and the mass of the CDPR with model uncertainties are assumed the same as those of the CDPR without model uncertainty, as shown in Table 4.9. The Jacobian of the CDPR used in the inverse dynamics equation in the hybrid DDPG strategy is calculated based on the

CDPR without model uncertainty. Thus, the inverse dynamics equation of the CDPR is no longer accurate in controlling the CDPR with model uncertainties due to the position errors of anchor points.

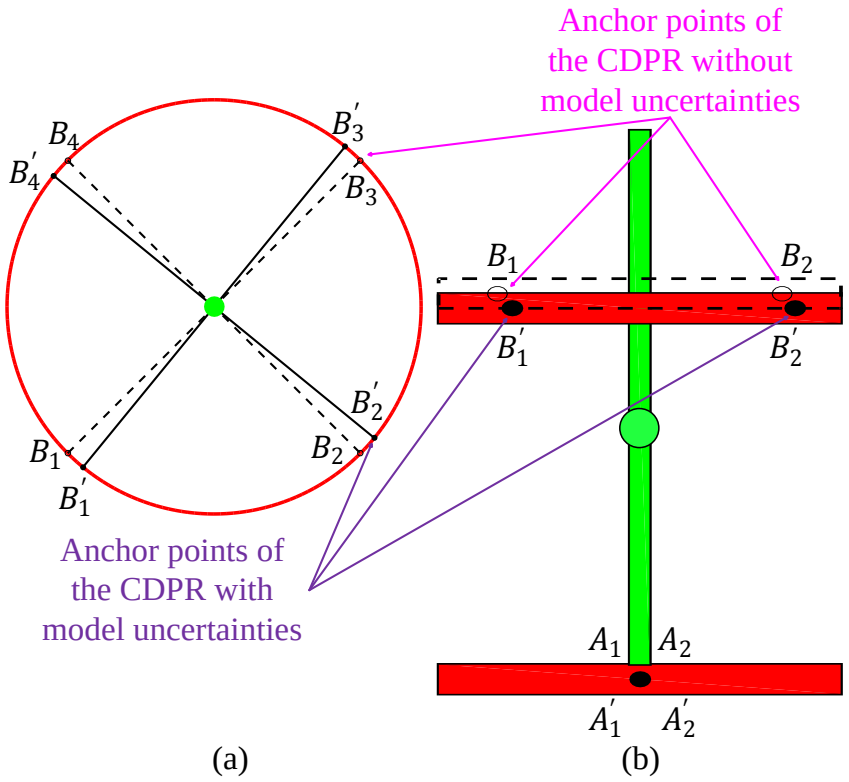


Figure 4.28 Position errors of anchor points of the CDPR:
a) top view; and b) side view

Table 4.14 Positions of anchor points of the CDPR with model uncertainties (unit: m)

Pair of anchor points	**Positions of anchor points on the base**	**Positions of anchor points on the end-effector**
$1'$	A_1': $[0.1, 0.0, -0.15]^T$	B_1': $[0.0774, -0.0633, 0.04]^T$
$2'$	A_2': $[0.1, 0.0, -0.15]^T$	B_2': $[0.0633, 0.0774, 0.04]^T$
$3'$	A_3': $[-0.1, 0.0, -0.15]^T$	B_3': $[-0.0774, 0.0633, 0.04]^T$
$4'$	A_4': $[-0.1, 0.0, -0.15]^T$	B_4': $[-0.0633, -0.0774, 0.04]^T$

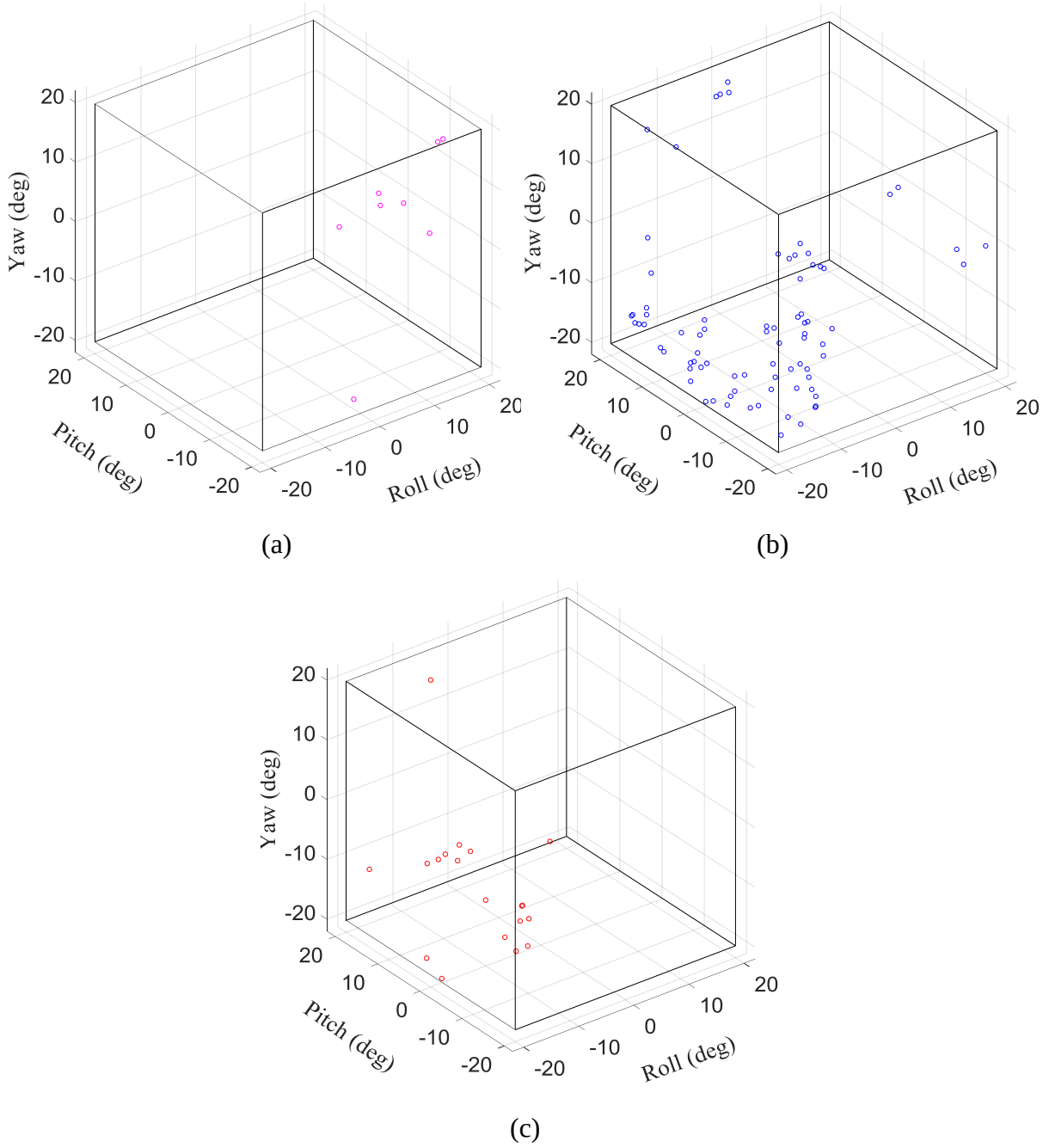
The pose-tracking test and the trajectory-tracking test are conducted on a CDPR with model uncertainties in this section. In the pose-tracking test, the target poses are the poses shown in Figure

4.24. The outcomes of the pose-tracking test are shown in Table 4.15. The control performance of the end-to-end DDPG strategy is affected by the model uncertainties. However, the control performance of the hybrid DDPG strategy is not affected much, even though the inverse dynamics equation of the CDPR is not accurate in this case. Therefore, the hybrid DDPG strategy is more robust to model uncertainties than the end-to-end DDPG strategy. Target poses that the CDPR fails to reach are shown in Figure 4.29. Target poses that the CDPR fails to reach are close to the boundary of the training workspace.

Table 4.15 Outcomes of the pose-tracking test with model uncertainties

Strategy	**Number of target poses the CDPR reaches successfully**	**Number of target poses the CDPR fails to reach**	**Success rate**
Hybrid DDPG strategy with the reward of r_t^w	992	8	99.2%
End-to-end DDPG strategy with the reward of $r_t^{\tau 1}$	921	79	92.1%
End-to-end DDPG strategy with the reward of $r_t^{\tau 2}$	981	19	98.1%

For the trajectory-tracking test, the CDPR with model uncertainties is controlled by the proposed strategies to track the target trajectory defined in (4.8). The trajectories and tracking errors of the CDPR with model uncertainties are shown in Figure 4.30 and Figure 4.31, respectively. The CDPR with model uncertainties can still be controlled by the proposed strategies to track the target trajectory. Namely, the proposed strategies are robust to certain model uncertainties. However, comparing Figure 4.27 and Figure 4.31, one can see that, controlled by the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$, the CDPR with model uncertainties has larger tracking errors (about 2.7 degrees) than the CDPR without model uncertainty (about 2.2 degrees). It means a decrease in control performance. Moreover, controlled by the hybrid DDPG strategy and the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$, the CDPR with model uncertainties has almost the same tracking errors as the CDPR without model uncertainty.



(a) (b)

(c)

Figure 4.29 Target poses fails to be reached by the CDPR with model uncertainties using: (a) the hybrid DDPG strategy; (b) the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$; and (c) the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$

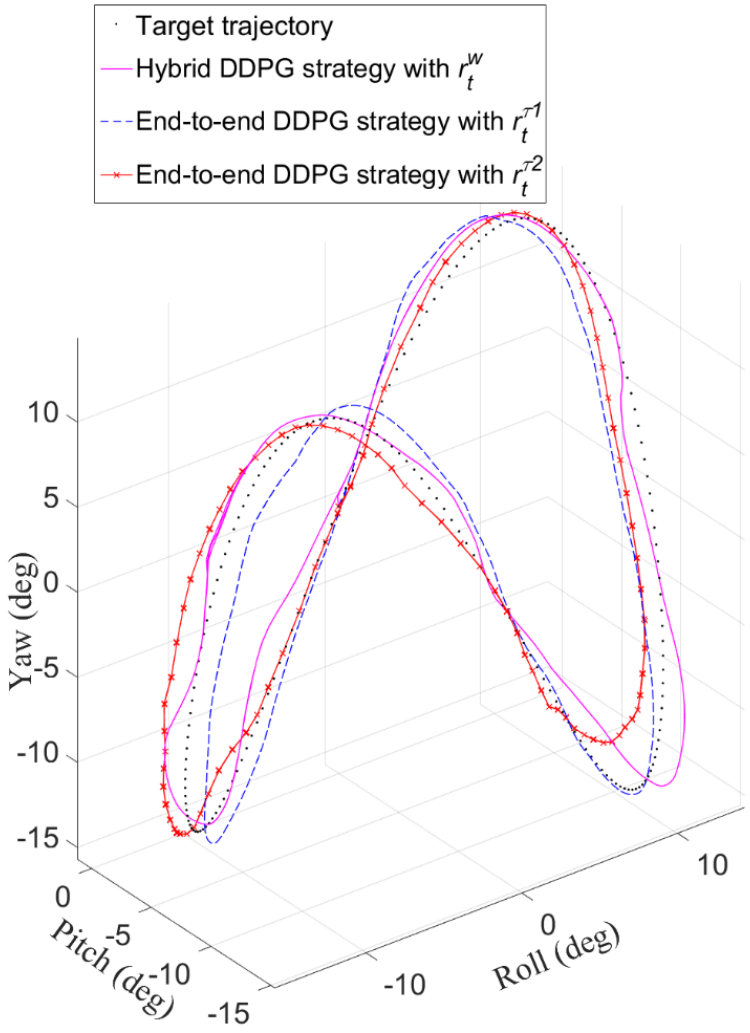


Figure 4.30 Trajectories of the CDPR with model uncertainties

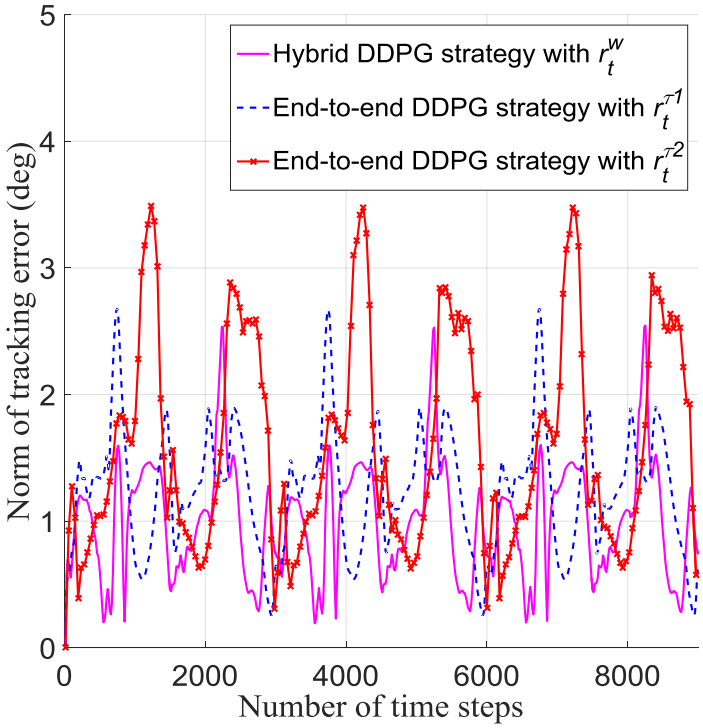


Figure 4.31 Tracking errors of the CDPR with model uncertainties

4.3.4 Conclusion

The study shows that the hybrid DDPG strategy learns faster than the end-to-end DDPG strategy in training. Both the hybrid DDPG strategy and the end-to-end DDPG strategy are robust to certain model uncertainties. However, the hybrid DDPG strategy is more robust to model uncertainties than the end-to-end DDPG strategy. Moreover, the end-to-end DDPG strategy can learn the optimal tension distribution of cables as well as the hybrid DDPG strategy calculates it from the inverse dynamics equation of the CDPR. The study demonstrates that, by taking advantages of both learning and non-learning-based approaches, the hybrid DDPG strategy provides an alternative to accomplish a robot manipulation task.

To control a CDPR in practice, the state variables of the proposed strategies can be measured, and the action variables can be implemented. State variables (i.e., the pose and velocity of the CDPR) can be measured by instruments such as IMUs (Aflakian, Safaryazdi, Tale Masouleh, & Kalhor, 2018). Action variables (i.e., cable tensions of the CDPR) can be delivered by direct-current

motors with proper voltage inputs (Niyetkaliyev et al., 2017). One should note that controlling a robot with DRL strategies may have safety issues in practice. For example, although controlling a robot with DRL strategies is feasible for non-safety-critical applications such as robotic palletizing, regulations may prevent the implementation of DRL strategies for safety-critical applications such as robotic surgery.

4.4 Safety Robustness of Learning Control Strategies

When RL policies are employed to control CDPRs for rehabilitation (Xiong & Diao, 2019), safety robustness of policies bears a concern because the failure of policies may injure trainees. When using an assistive device for rehabilitation, the safety of the trainee always has a higher priority than the success of a training maneuver assisted by the assistive device. Thus, one would like to evaluate how safe a policy is in manipulating an assistive device when subject to uncertainties and perturbations.

This section aims to demonstrate how to use safety robustness to evaluate the robustness policies, rather than how to improve the safety robustness of policies or how to better shape reward functions. Three policies trained by DDPG with three different reward functions to control a CDPR to assist human joints are used in this section. For the section, the Gazebo simulator (Koenig & Howard, 2004) is used to conduct simulations.

The CDPR used in this study has three rotational degrees of freedom and four cables, as shown in Figure 4.32. The problem to be solved by RL is to control the orientation of the end-effector of the CDPR via cable tensions. Cable tensions are assumed to range from 0 N to 20 N in this study. The orientation of the end-effector of the CDPR is represented by three Euler angles $[\phi, \theta, \psi]$ with a ψ-θ-ϕ sequence.

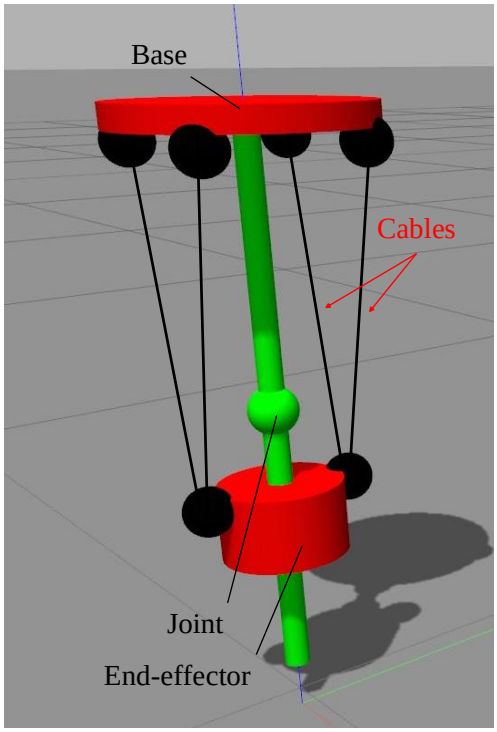


Figure 4.32 The model of a CDPR in the Gazebo simulator

4.4.1 Learning Setup

For the study, the state space is 9-dimensional and the action space is 4-dimensional. Namely, $S \in \mathbb{R}^9$ and $A \in \mathbb{R}^4$. State variables at time step t can be expressed as

$$\boldsymbol{x}_t = [\boldsymbol{X}, \dot{\boldsymbol{X}}, \Delta\boldsymbol{X}] \tag{4.9}$$

where $\boldsymbol{X} = [\phi, \theta, \psi]$ is the orientation of the end-effector of the CDPR at time step t. $\dot{\boldsymbol{X}} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$ is the angular velocity of the CDPR at time step t. $\Delta\boldsymbol{X}$ is the difference between the target orientation of the end-effector of the CDPR, denoted as $\boldsymbol{X}^* = [\phi^*, \theta^*, \psi^*]$, and the orientation of the end-effector of the CDPR at time step t. Thus, $\Delta\boldsymbol{X}$ can be expressed as

$$\Delta\boldsymbol{X} = \boldsymbol{X}^* - \boldsymbol{X} = [\phi - \phi^*, \theta - \theta^*, \psi - \psi^*] \tag{4.10}$$

$\boldsymbol{x}_t$ is a 9-dimenionsal state vector. Action variables at time step t can be expressed as

$$\boldsymbol{a}_t = [\tau_1, \tau_2, \tau_3, \tau_4] \tag{4.11}$$

where τ_i $(i = 1, 2, 3, 4)$ is the cable tension in the ith cable. $\boldsymbol{u}_t$ is a 4-dimensional action vector.

In the simulations, policies are trained by DDPG. Three reward functions are designed to guide an agent to learn three different policies. The three reward functions are expressed as

$$r_1 = -[(\sqrt{|\phi - \phi^*|} + \sqrt{|\theta - \theta^*|} + \sqrt{|\psi - \psi^*|}) + 0.01(\dot{\phi}^2 + \dot{\theta}^2 + \dot{\psi}^2) + 0.0001 \sum \tau_i^2] \quad (4.12)$$

$$r_2 = -[(|\phi - \phi^*| + |\theta - \theta^*| + |\psi - \psi^*|) + 0.01(\dot{\phi}^2 + \dot{\theta}^2 + \dot{\psi}^2) + 0.0001 \sum \tau_i^2] \quad (4.13)$$

$$r_3 = -[(\phi - \phi^*)^2 + (\theta - \theta^*)^2 + (\psi - \psi^*)^2 + 0.0001 \sum \tau_i^2] \quad (4.14)$$

All three reward functions can guide an agent to learn policies that move the end-effector of the CDPR to a target orientation. With r_1, r_2, and r_3, the policies π_1, π_2, and π_3 are learned by the agent accordingly. The four networks of the DDPG are fully connected neural networks with two hidden layers. The architecture of the four networks of the DDPG is shown in Table 4.16. Hyper-parameters used in training are shown in Table 4.17. The learning rates of the critic network and the critic-target network change from 0.0001 to 0.00005 if the training loss is less than 0.001.

Table 4.16 Architecture of the four networks of the DDPG

	Actor and actor-target networks	**Critic and critic-target networks**
Number of inputs	9	9+4
Activation function 1	ReLU	ReLU
Number of units in layer 1	64	64
Activation function 2	ReLU	ReLU
Number of units in layer 2	64	64
Activation function 3	tanh	None
Number of outputs	4	1

Table 4.17 Hyper-parameters in training

Hyper-Parameter	**Value**
Maximum number of time steps for each episode	1000
Learning rate of the actor and actor-target networks	0.001
Learning rate of the critic and critic-target networks	$0.0001 \rightarrow 0.00005$
Discount factor γ	0.95
Update rate (target) τ	0.001
Size of the experience replay buffer	10000
Size of mini-batch	1024
Maximum magnitude of random exploration noise	$\frac{\text{range of an action}}{\text{number of episodes} + 2}$

4.4.2 Safety Robustness Analysis

Safety robustness is applied to evaluate the robustness of the three policies (i.e., π_1, π_2, and π_3). In the simulations, the number of episodes is M = 1000 and the number of steps is T = 100. Δ_s^{max} and Δ_a^{max} are set to be the same, represented by Δ^{max}. Δ^{max} ranges from 0 to 0.5 with a step size of 0.1. Without hyperextending a joint, safe states are defined as states that satisfy

$$|\phi| < 30 \text{ deg},\ |\theta| < 30 \text{ deg},\ |\psi| < 30 \text{ deg} \tag{4.15}$$

For each episode, a random target orientation satisfying

$$|\phi| < 15 \text{ deg},\ |\theta| < 15 \text{ deg},\ |\psi| < 15 \text{ deg} \tag{4.16}$$

is set for policies. In an episode, the end-effector of the CDPR is initialized to the orientation $[0, 0, 0]$ then controlled by a policy to a randomly generated target orientation $[\phi^*, \theta^*, \psi^*]$. Safety robustness with 95% confidence interval (based on 1000 tests) of the three policies is plotted in Figure 4.33. Table 4.18 lists the numerical values of the safety robustness of the three policies.

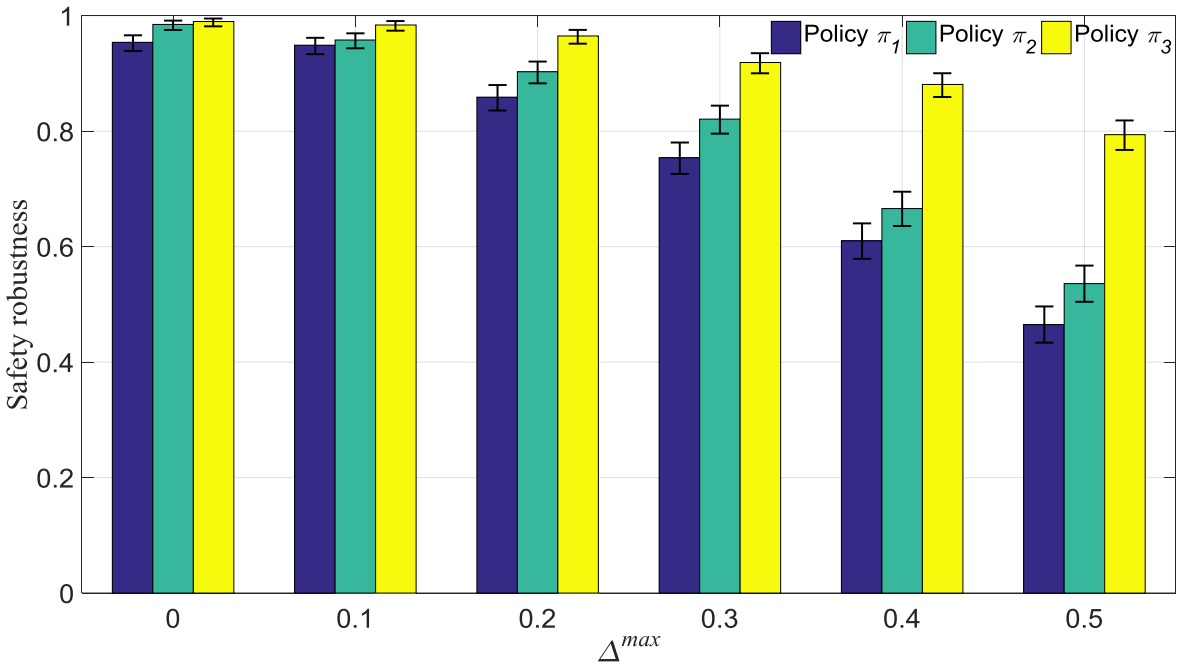


Figure 4.33 Safety robustness of the three policies

Table 4.18 Safety robustness of the three policies

$\boldsymbol{\Delta^{max}}$	**Safety robustness of $\boldsymbol{\pi_1}$**	**Safety robustness of $\boldsymbol{\pi_2}$**	**Safety robustness of $\boldsymbol{\pi_3}$**
0.0	0.954	0.985	0.990
0.1	0.949	0.958	0.984
0.2	0.859	0.903	0.965
0.3	0.754	0.821	0.919
0.4	0.610	0.666	0.881
0.5	0.465	0.536	0.794

As shown in Figure 4.33, the safety robustness of all three policies decreases with the increase of the magnitude of uncertainties or perturbations. This suggests that increasing uncertainties or perturbations degrade the safety robustness of policies. In other words, should uncertainties and perturbations be attenuated, the safety robustness of policies would be improved. Another observation is that policies learned with the same DDPG but different reward functions have different safety robustness. Policy π_3 has better safety robustness than policies π_1 and π_2. This means that policy π_3 is safer than policies π_1 and π_2 when subject to uncertainties or perturbations.

The safety robustness of policies to different magnitudes of uncertainties or perturbations is approximated statistically. Based on the safety robustness, one can tell how safe the policies are when subject to uncertainties or perturbations. Moreover, one can find that policy π_3 has the best safety robustness among the three policies. In other words, one should select policy π_3 to control the CDPR to mitigate safety concerns.

4.4.3 Conclusion

Safety robustness reflects a policy's ability to maintain safe state when subject to uncertainties or perturbations. In section 4.4, simulations are conducted to demonstrate how to approximate safety robustness of policies. Simulation results show that safety robustness can be used to evaluate the robustness of policies and select policies from the perspective of safety.

4.5 Summary

For chapter four, simulations are conducted to demonstrate the workspace isotropy and the moment efficiency of CDADs. Workspace isotropy and moment efficiency can be used for the optimization of CDADs. Moreover, based on simulation conducted in Gazebo, the effectiveness of the PID–FNN control strategy and DDPG-based strategies are verified. The safety robustness of DDPG-based strategies is analyzed as well.

CHAPTER 5. SUMMARY

Learning control strategies are developed in this dissertation to make a CDAD intelligent in assisting a human joint. The summary of the development of learning control strategies for CDADs and the recommendations for future work of learning control strategies for CDADs are demonstrated in this chapter.

5.1 Summary

The dissertation has the following contributions to the communities of assistive devices, robotics, control, and reinforcement learning:

- The dissertation addresses the isotropy of a workspace in terms of its geometric shape. The dissertation proposes three novel workspace isotropy indices - TWII, RWII, and EWII - for workspace isotropy analysis in terms of the geometric shape of the workspace. All three indices are mathematically defined. Simulation results show that the proposed workspace isotropy indices reflect how isotropic the geometric shape of a workspace is.
- The dissertation studies how a CDAD with three DOFs and four cables exerts a joint force on a general three-DOF human joint. An index to evaluate the efficiency of a CDAD in providing an assistant moment is proposed. The moment efficiency index provides a guideline for the design of CDADs.
- A learning-based control framework consisting of a robust controller and an ANN in series is proposed. As an example, a PID–FNN control strategy is developed to demonstrate how the proposed control framework works. The PID–FNN control strategy includes a PID controller and a feedforward neural network. Simulation results show that the PID–FNN control strategy can successfully control a CDPR with four cables, three degrees of freedom, and unknown Jacobian.
- For the study, an end-to-end DRL strategy and a hybrid DRL strategy are developed and compared in controlling a cable-driven parallel robot. The study shows that, by integrating DRL with non-learning-based approaches, the hybrid DRL strategy learns faster and is more robust to model uncertainties than the end-to-end DRL strategy. Specifically, the hybrid DRL strategy converges in 70 episodes while the end-to-end DRL

strategies converge in more than 100 episodes. When the same model uncertainty is applied, the success rate of the hybrid DRL strategy decreases from 99.3% to 99.2%, while the success rate of the end-to-end DRL strategies with $r_t^{\tau 1}$ and $r_t^{\tau 2}$ decrease from 97.8% to 92.1% and 99.5% to 98.1%, respectively. The study demonstrates that, by taking advantage of both learning and non-learning-based approaches, the hybrid DRL strategy provides an alternative to accomplish a joint assisting task.

- Inspired by the system architecture of robust control and the concept of safe state in safe RL, the dissertation proposes the concept of safety robustness to evaluate the robustness of policies from the perspective of safety. Safety robustness reflects the ability of a policy to maintain safe state when subject to uncertainties and perturbations. Simulations are conducted to demonstrate how to analyze safety robustness to evaluate the robustness of policies.

5.2 Recommendations

Two important research issues to further enrich the developed learning control strategies for CDADs are discussed in section 5.2.

- The proposed learning control strategies will be further improved based on more simulations and experiments. For the study, simulations are conducted aiming to validate the effectiveness of the proposed learning control strategies. However, the proposed learning control strategies have not been optimized based on the parameters of a real human joint. Thus, the proposed learning control strategies can be further improved based on simulations and experiments with the parameters of a real human joint.
- Detailed guidelines of the learning control strategies for CDADs will be published to improve the ease of use. Learning control strategies have been proposed and validated in this study. However, some details about how to use the proposed learning control strategies are not fully discussed. Therefore, more details, such as the optimal choice of ANNs, will be discussed in the future.

LIST OF REFERENCES

Aflakian, A., Safaryazdi, A., Tale Masouleh, M., & Kalhor, A. (2018). Experimental study on the kinematic control of a cable suspended parallel robot for object tracking purpose. *Mechatronics*, *50*, 160–176. https://doi.org/https://doi.org/10.1016/j.mechatronics.2018.02.005

Alamdari, A., & Krovi, V. (2015a). Modeling and control of a novel home-based cable-driven parallel platform robot: PACER. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 6330–6335). https://doi.org/10.1109/IROS.2015.7354281

Alamdari, A., & Krovi, V. (2015b). Robotic physical exercise and system (ROPES): A cable-driven robotic rehabilitation system for lower-extremity motor therapy. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers.

Alikhani, A., & Vali, M. (2011). Modeling and robust control of a new large scale suspended cable-driven robot under input constraint. In *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)* (pp. 238–243). https://doi.org/10.1109/URAI.2011.6145969

Alp, A. B., & Agrawal, S. K. (2002). Cable suspended robots: feedback controllers with positive inputs. In *American Control Conference (ACC)* (Vol. 1, pp. 815–820 vol.1). https://doi.org/10.1109/ACC.2002.1024915

Anastassiou, G. A. (2011). Multivariate sigmoidal neural network approximation. *Neural Networks*, *24*(4), 378–386.

Asl, H. J., & Janabi-Sharifi, F. (2017). Adaptive neural network control of cable-driven parallel robots with input saturation. *Engineering Applications of Artificial Intelligence*, *65*, 252–260.

Azadi, M., Behzadipour, S., & Faulkner, G. (2009). Antagonistic variable stiffness elements. *Mechanism and Machine Theory*, *44*(9), 1746–1758. https://doi.org/https://doi.org/10.1016/j.mechmachtheory.2009.03.002

Babaghasabha, R., Khosravi, M. A., & Taghirad, H. D. (2015). Adaptive robust control of fully-constrained cable driven parallel robots. *Mechatronics*, *25*, 27–36. https://doi.org/https://doi.org/10.1016/j.mechatronics.2014.11.005

Babaiasl, M., Mahdioun, S. H., Jaryani, P., & Yazdani, M. (2016). A review of technological and clinical aspects of robot-aided rehabilitation of upper-extremity after stroke. *Disability and Rehabilitation: Assistive Technology*, *11*(4), 263–280.

Bae, J., Siviy, C., Rouleau, M., Menard, N., O'Donnell, K., Geliana, I., … Walsh, C. J. (2018). A Lightweight and Efficient Portable Soft Exosuit for Paretic Ankle Assistance in Walking After Stroke. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2820–2827). https://doi.org/10.1109/ICRA.2018.8461046

Balasubramanian, S., Klein, J., & Burdet, E. (2010). Robot-assisted rehabilitation of hand function. *Current Opinion in Neurology*, *23*(6), 661–670.

Ball, S., Brown, I., & Scott, S. (2007). Designing a Robotic Exoskeleton for Shoulder Complex Rehabilitation. *CMBES Proceedings*, *30*(1 SE-Academic).

Ball, S. J., Brown, I. E., & Scott, S. H. (2007). MEDARM: a rehabilitation robot with 5DOF at the shoulder complex. In *IEEE/ASME international conference on advanced intelligent mechatronics* (pp. 1–6). https://doi.org/10.1109/AIM.2007.4412446

Barreca, S., Wolf, S. L., Fasoli, S., & Bohannon, R. (2003). Treatment interventions for the paretic upper limb of stroke survivors: a critical review. *Neurorehabilitation and Neural Repair*, *17*(4), 220–226.

Barrette, G., & Gosselin, C. M. (2005). Determination of the Dynamic Workspace of Cable-Driven Planar Parallel Mechanisms. *Journal of Mechanical Design*, *127*(March 2005), 242. https://doi.org/10.1115/1.1830045

Basteris, A., Nijenhuis, S. M., Stienen, A. H. A., Buurke, J. H., Prange, G. B., & Amirabdollahian, F. (2014). Training modalities in robot-mediated upper limb rehabilitation in stroke: a framework for classification based on a systematic review. *Journal of Neuroengineering and Rehabilitation*, *11*(1), 111.

Begey, J., Cuvillon, L., Lesellier, M., Gouttefarde, M., & Gangloff, J. (2019). Dynamic Control of Parallel Robots Driven by Flexible Cables and Actuated by Position-Controlled Winches. *IEEE Transactions on Robotics*, *35*(1), 286–293. https://doi.org/10.1109/TRO.2018.2875415

Behzadipour, S., & Khajepour, A. (2005). Stiffness of Cable-based Parallel Manipulators With Application to Stability Analysis. *Journal of Mechanical Design*, *128*(1), 303–310. Retrieved from http://dx.doi.org/10.1115/1.2114890

Behzadipour, S., & Sohi, M. A. (2007). Antagonistic stiffness in cable-driven mechanisms. In *IFToMM World Congress, Besançon, France, In*.

Belda-Lois, J.-M., Mena-del Horno, S., Bermejo-Bosch, I., Moreno, J. C., Pons, J. L., Farina, D., … Rea, M. (2011). Rehabilitation of gait after stroke: a review towards a top-down approach. *Journal of NeuroEngineering and Rehabilitation*, *8*(1), 66. https://doi.org/10.1186/1743-0003-8-66

Bhattacharyya, S. P. (2017). Robust control under parametric uncertainty: An overview and recent results. *Annual Reviews in Control*, *44*, 45–77. https://doi.org/https://doi.org/10.1016/j.arcontrol.2017.05.001

Brackbill, E. A., Mao, Y., Agrawal, S. K., Annapragada, M., & Dubey, V. N. (2009). Dynamics and control of a 4-dof wearable cable-driven upper arm exoskeleton. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2300–2305). https://doi.org/10.1109/ROBOT.2009.5152545

Cao, J., Xie, S. Q., Das, R., & Zhu, G. L. (2014). Control strategies for effective robot assisted gait rehabilitation: the state of art and future prospects. *Medical Engineering & Physics*, *36*(12), 1555–1566.

Caverly, R. J., & Forbes, J. R. (2014). Dynamic modeling and noncollocated control of a flexible planar cable-driven manipulator. *IEEE Transactions on Robotics*, *30*(6), 1386–1397.

Chen, G., Chan, C. K., Guo, Z., & Yu, H. (2013). A review of lower extremity assistive robotic exoskeletons in rehabilitation therapy. *Critical Reviews™ in Biomedical Engineering*, *41*(4–5).

Chen, W., Cui, X., Zhang, J., & Wang, J. (2015). A cable-driven wrist robotic rehabilitator using a novel torque-field controller for human motion training. *Review of Scientific Instruments*, *86*(6), 0–14. https://doi.org/10.1063/1.4923089

Chen, Z., Cao, F., & Hu, J. (2015). Approximation by network operators with logistic activation functions. *Applied Mathematics and Computation*, *256*, 565–571. https://doi.org/https://doi.org/10.1016/j.amc.2015.01.049

Costarelli, D., & Spigler, R. (2013). Approximation results for neural network operators activated by sigmoidal functions. *Neural Networks*, *44*, 101–106.

Craig, J. J. (2009). *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India.

Cui, X., Chen, W., Jin, X., & Agrawal, S. K. (2017). Design of a 7-dof cable-driven arm exoskeleton (carex-7) and a controller for dexterous motion training or assistance. *IEEE/ASME Transactions on Mechatronics*, *22*(1), 161–172.

Cui, X., Chen, W., Zhang, J., & Wang, J. (2015). Note: Model-based identification method of a cable-driven wearable device for arm rehabilitation. *Review of Scientific Instruments*, *86*(9), 096107. https://doi.org/10.1063/1.4931577

Da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., & dos Reis Alves, S. F. (2017). Artificial neural networks. *Cham: Springer International Publishing*.

Dhir, N., Dallali, H., Ficanha, E. M., Ribeiro, G. A., & Rastgaar, M. (2018). Locomotion Envelopes for Adaptive Control of Powered Ankle Prostheses. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1488–1495). https://doi.org/10.1109/ICRA.2018.8460929

Diao, X., & Ma, O. (2007). A method of verifying force-closure condition for general cable manipulators with seven cables. *Mechanism and Machine Theory*, *42*(12), 1563–1576. https://doi.org/10.1016/j.mechmachtheory.2007.06.008

Diao, X., & Ma, O. (2008). Workspace Determination of General 6-d.o.f. Cable Manipulators. *Advanced Robotics*, *22*(2–3), 261–278. https://doi.org/10.1163/156855308x292574

Diao, X., & Ma, O. (2009). Vibration analysis of cable-driven parallel manipulators. *Multibody System Dynamics*, *21*(4), 347–360. https://doi.org/10.1007/s11044-008-9144-0

Dobkin, B. H. (2004). Strategies for stroke rehabilitation. *The Lancet Neurology*, *3*(9), 528–536.

Doyle, J. C., Wall, J. E., & Stein, G. (1982). Performance and robustness analysis for structured uncertainty. In *1982 21st IEEE Conference on Decision and Control* (pp. 629–636).

Du, J., & Agrawal, S. K. (2015). Dynamic Modeling of Cable-Driven Parallel Manipulators With Distributed Mass Flexible Cables. *Journal of Vibration and Acoustics*, *137*(2), 21020. https://doi.org/10.1115/1.4029486

Duan, Y., Chen, X., Houthooft, R., Schulman, J., & Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning* (pp. 1329–1338).

Dullerud, G. E., & Paganini, F. (2013). *A course in robust control theory: a convex approach* (Vol. 36). Springer Science & Business Media.

Esmaeili, M., Gamage, K., Tan, E., & Campolo, D. (2011). Ergonomic considerations for anthropomorphic wrist exoskeletons: A simulation study on the effects of joint misalignment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4905–4910). IEEE.

Eugene, L., Kevin, W., & Howe, D. (2013). Robust and adaptive control with aerospace applications. Springer London.

Fang, S., Franitza, D., Torlo, M., Bekes, F., & Hiller, M. (2004). Motion control of a tendon-based parallel manipulator using optimal tension distribution. *IEEE/ASME Transactions On Mechatronics*, *9*(3), 561–568. https://doi.org/https://doi.org/10.1109/TMECH.2004.835336

Fattah, A., & Ghasemi, A. M. H. (n.d.). Isotropic Design of Spatial Parallel Manipulators. *The International Journal of Robotics Research*.

Fazekas, G., Horvath, M., Troznai, T., & Toth, A. (2007). Robot-mediated upper limb physiotherapy for patients with spastic hemiparesis: a preliminary study. *Journal of Rehabilitation Medicine*, *39*(7), 580–582.

Feki, M. S., Chaari, F., Abbes, M. S., Viadero, F., Fdez. Del Rincon, A., & Haddar, M. (2013). Hybrid Position/Force Control of a Cable-Driven Parallel Robot with Experimental Evaluation. *Mechanisms and Machine Science*, *7*, 311–318. https://doi.org/10.1007/978-94-007-4902-3

Fernandez-Gauna, B., Fernandez-Gamiz, U., & Grana, M. (2017). Variable speed wind turbine controller adaptation by reinforcement learning. *Integrated Computer-Aided Engineering*, *24*(1), 27–39. Retrieved from doi:10.3233/ICA-160531

Funahashi, K., & Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, *6*(6), 801–806.

Galiana, I., Hammond, F. L., Howe, R. D., & Popovic, M. B. (2012). Wearable soft robotic device for post-stroke shoulder rehabilitation: Identifying misalignments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 317–322). https://doi.org/10.1109/IROS.2012.6385786

Gallina, P., & Rosati, G. (2002). Manipulability of a planar wire driven haptic device. *Mechanism and Machine Theory*, *37*(2), 215–228. https://doi.org/10.1016/S0094-114X(01)00076-3

Gaponov, I., Popov, D., Lee, S. J., & Ryu, J.-H. (2017). Auxilio: A portable cable-driven exosuit for upper extremity assistance. *International Journal of Control, Automation and Systems*, *15*(1), 73–84. https://doi.org/10.1007/s12555-016-0487-7

Garcia, J., & Fernández, F. (2012). Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, *45*, 515–564.

Garcıa, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, *16*(1), 1437–1480.

Geibel, P., & Wysotzki, F. (2005). Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, *24*, 81–108.

Ghasemi, A., Eghtesad, M., & Farid, M. (2010). Neural Network Solution for Forward Kinematics Problem of Cable Robots. *Journal of Intelligent & Robotic Systems*, *60*(2), 201–215. https://doi.org/10.1007/s10846-010-9421-z

Giorelli, M., Renda, F., Calisti, M., Arienti, A., Ferri, G., & Laschi, C. (2015). Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature. *IEEE Transactions on Robotics*, *31*(4), 823–834.

Giorelli, M., Renda, F., Ferri, G., & Laschi, C. (2013). A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (pp. 5033–5039). IEEE.

Golemo, F., Taïga, A. A., Oudeyer, P.-Y., & Courville, A. (2018). Sim-to-Real Transfer with Neural-Augmented Robot Simulation. *2nd Conference on Robot Learning (CoRL18)*, (CoRL).

Gopura, R. A. R. C., Bandara, D. S. V, Kiguchi, K., & Mann, G. K. I. (2016). Developments in hardware systems of active upper-limb exoskeleton robots: A review. *Robotics and Autonomous Systems*, *75*, 203–220. https://doi.org/10.1016/j.robot.2015.10.001

Gosselin, C., & Angeles, J. (1989). The Optimum Kinematic Design of a Spherical Three-Degree-of-Freedom Parallel Manipulator. *Journal of Mechanisms Transmissions and Automation in Design*, *111*(2), 202. https://doi.org/10.1115/1.3258984

Gosselin, C., & Angeles, J. (1991). A global performance index for the kinematic optimization of robotic manipulators. *Journal of Mechanical Design*, *113*(3), 220–226.

Gosselin, C., & Lavoie, E. (1993). On the Kinematic Design of Spherical Three-Degree-of-Freedom Parallel Manipulators. *The International Journal of Robotics Research*, *12*(4), 394–402. https://doi.org/10.1177/027836499301200406

Gouttefarde, M., & Gosselin, C. M. (2004). On the properties and the determination of the wrench-closure workspace of planar parallel cable-driven mechanisms. In *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 337–346). American Society of Mechanical Engineers.

Gu, S., Lillicrap, T., Sutskever, I., & Levine, S. (2016). Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning* (pp. 2829–2838).

Gu, Z., Jia, Z., & Choset, H. (n.d.). Adversary a3c for robust reinforcement learning.

Haykin, S. S., Haykin, S. S., Haykin, S. S., Elektroingenieur, K., & Haykin, S. S. (2009). *Neural networks and learning machines* (Vol. 3). Pearson education Upper Saddle River.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018). Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Heo, P., Gu, G. M., Lee, S., Rhee, K., & Kim, J. (2012). Current hand exoskeleton technologies for rehabilitation and assistive engineering. *International Journal of Precision Engineering and Manufacturing*, *13*(5), 807–824.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, *18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Huang, X., Yan, Y., Zhou, Y., & Xu, H. (2015). Output feedback sliding mode control of Lorentz-augmented spacecraft hovering using neural networks. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, *229*(10), 939–948. https://doi.org/10.1177/0959651815599675

Huo, W., Mohammed, S., Moreno, J. C., & Amirat, Y. (2016). Lower Limb Wearable Robots for Assistance and Rehabilitation: A State of the Art. *IEEE Systems Journal*, *10*(3), 1068–1081. https://doi.org/10.1109/JSYST.2014.2351491

Hussain, S. (2014). State-of-the-art robotic gait rehabilitation orthoses: design and control aspects. *NeuroRehabilitation*, *35*(4), 701–709.

Ikuta, K., Ishii, H., & Nokata, M. (2003). Safety evaluation method of design and control for human-care robots. *The International Journal of Robotics Research*, *22*(5), 281–297.

In, H., Cho, K., Kim, K., & Lee, B. (2011). Jointless structure and under-actuation mechanism for compact hand exoskeleton. In *International Conference on Rehabilitation Robotics (ICORR)* (pp. 1–6). https://doi.org/10.1109/ICORR.2011.5975394

Jakobi, N., Husbands, P., & Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life* (pp. 704–720). Springer.

Jamshidifar, H., Khosravani, S., Fidan, B., & Khajepour, A. (2018). Vibration Decoupled Modeling and Robust Control of Redundant Cable-Driven Parallel Robots. *IEEE/ASME Transactions on Mechatronics*, *23*(2), 690–701. https://doi.org/10.1109/TMECH.2018.2793578

Jamwal, P., Hussain, S., & Xie, S. (2013). Review on design and control aspects of ankle rehabilitation robots. *Disability and Rehabilitation. Assistive Technology*, *10*(2), 1–9. https://doi.org/10.3109/17483107.2013.866986

Jamwal, P., Xie, S., & Aw, K. (2009). Kinematic design optimization of a parallel ankle rehabilitation robot using modified genetic algorithm. *Robotics and Autonomous Systems*, *57*(10), 1018–1027.

Jarrassé, N., Proietti, T., Crocher, V., Robertson, J., Sahbani, A., Morel, G., & Roby-Brami, A. (2014). Robotic exoskeletons: a perspective for the rehabilitation of arm coordination in stroke patients. *Frontiers in Human Neuroscience*, *8*, 947.

Jeong, U., In, H.-K., & Cho, K.-J. (2013). Implementation of various control algorithms for hand rehabilitation exercise using wearable robotic hand. *Intelligent Service Robotics*, *6*(4), 181–189.

Jin, L., Li, S., Yu, J., & He, J. (2018). Robot manipulator control using neural networks: A survey. *Neurocomputing*, *285*, 23–34. https://doi.org/https://doi.org/10.1016/j.neucom.2018.01.002

Jin, L., Zhang, Y., & Li, S. (2016). Integration-Enhanced Zhang Neural Network for Real-Time-Varying Matrix Inversion in the Presence of Various Kinds of Noises. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(12), 2615–2627. https://doi.org/10.1109/TNNLS.2015.2497715

Jin, X., Cui, X., & Agrawal, S. K. (2015). Design of a cable-driven active leg exoskeleton (C-ALEX) and gait training experiments with human subjects. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5578–5583). https://doi.org/10.1109/ICRA.2015.7139979

Kang, B. B., In, H., & Cho, K. (2012). Force transmission in joint-less tendon driven wearable robotic hand. In *International Conference on Control, Automation and Systems* (pp. 1853–1858).

Kesner, S. B., Jentoft, L., Hammond, F. L., Howe, R. D., & Popovic, M. (2011). Design considerations for an active soft orthotic system for shoulder rehabilitation. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 8130–8134). https://doi.org/10.1109/IEMBS.2011.6092006

Khosravi, M., & Taghirad, H. (2014a). Dynamic Modeling and Control of Parallel Robots With Elastic Cables: Singular Perturbation Approach. *IEEE Transactions on Robotics*, *30*(3), 694–704.

Khosravi, M., & Taghirad, H. (2014b). Robust PID control of fully-constrained cable driven parallel robots. *Mechatronics*, *24*(2), 87–97. https://doi.org/https://doi.org/10.1016/j.mechatronics.2013.12.001

Kim, D. H., & Park, H. (2018). Cable Actuated Dexterous (CADEX) Glove for Effective Rehabilitation of the Hand for Patients with Neurological diseases. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2305–2310). https://doi.org/10.1109/IROS.2018.8594336

Kim, J., Heimgartner, R., Lee, G., Karavas, N., Perry, D., Ryan, D. L., … Walsh, C. J. (2018). Autonomous and Portable Soft Exosuit for Hip Extension Assistance with Online Walking and Running Detection Algorithm. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1–8). https://doi.org/10.1109/ICRA.2018.8460474

Kim, S., Kim, H., & Moon, B. (2005). Local and global isotropy of caster wheeled omnidirectional mobile robot. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (pp. 3446–3451). IEEE.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*.

Kino, H., Yahiro, T., Takemura, F., & Morizono, T. (2007). Robust PD control using adaptive compensation for completely restrained parallel-wire driven robots: Translational systems using the minimum number of wires under zero-gravity condition. *IEEE Transactions on Robotics*, *23*(4), 803–812.

Kino, H., Yoshitake, T., Wada, R., Tahara, K., & Tsuda, K. (2018). 3-DOF planar parallel-wire driven robot with an active balancer and its model-based adaptive control. *Advanced Robotics*, *32*(14), 766–777. https://doi.org/10.1080/01691864.2018.1493397

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, *32*(11), 1238–1274. https://doi.org/https://doi.org/10.1177%2F0278364913495721

Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vol. 3, pp. 2149–2154 vol.3). https://doi.org/10.1109/IROS.2004.1389727

Korayem, M. H., Yousefzadeh, M., & Beyranvand, B. (2017). Dynamics and Control of a 6-dof Cable-driven Parallel Robot with Visco-elastic Cables in Presence of Measurement Noise. *Journal of Intelligent & Robotic Systems*, *88*(1), 73–95. https://doi.org/10.1007/s10846-017-0546-1

Kwakkel, G., Kollen, B. J., & Krebs, H. I. (2008). Effects of robot-assisted therapy on upper limb recovery after stroke: a systematic review. *Neurorehabilitation and Neural Repair*, *22*(2), 111–121.

Lamaury, J., Gouttefarde, M., Chemori, A., & Herve, P.-E. (2013). Dual-space adaptive control of redundantly actuated cable-driven parallel robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4879–4886). IEEE.

Lee, S., Karavas, N., Quinlivan, B. T., LouiseRyan, D., Perry, D., Eckert-Erdheim, A., … Walsh, C. J. (2018). Autonomous Multi-Joint Soft Exosuit for Assistance with Walking Overground. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2812–2819). https://doi.org/10.1109/ICRA.2018.8460972

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, *17*(1), 1334–1373.

Li, F., Jiang, Q., Zhang, S., Wei, M., & Song, R. (2019). Robot Skill Acquisition in Assembly Process using Deep Reinforcement Learning. *Neurocomputing*. Retrieved from https://doi.org/10.1016/j.neucom.2019.01.087

Li, J., Wang, S., Wang, J., Zheng, R., Zhang, Y., & Chen, Z. (2012). Development of a hand exoskeleton system for index finger rehabilitation. *Chinese Journal of Mechanical Engineering*, *25*(2), 223–233.

Li, Y. (2017). Deep reinforcement learning: An overview. *ArXiv Preprint ArXiv:1701.07274*.

Li, Y., & Wang, Q. (2018). Adaptive neural finite-time trajectory tracking control of hydraulic excavators. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, *232*(7), 909–925. https://doi.org/10.1177/0959651818767770

Li, Z., Chen, W., Zhang, J., & Bai, S. (2017). Design and control of a 4-DOF cable-driven arm rehabilitation robot (CARR-4). In *IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)* (pp. 581–586). https://doi.org/10.1109/ICCIS.2017.8274842

Li, Z., Xue, S., Lin, W., & Tong, M. (2018). Training a robust reinforcement learning controller for the uncertain system based on policy gradient method. *Neurocomputing*, *316*, 313–321. Retrieved from https://doi.org/10.1016/j.neucom.2018.08.007

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., … Wierstra, D. (2015). Continuous control with deep reinforcement learning. *ArXiv Preprint ArXiv:1509.02971*.

Lim, W., Yang, G., Yeo, S., & Mustafa, S. (2011). A generic force-closure analysis algorithm for cable-driven parallel manipulators. *Mechanism and Machine Theory*, *46*(9), 1265–1275. https://doi.org/10.1016/j.mechmachtheory.2011.04.006

Lin, D., Wang, X., Nian, F., & Zhang, Y. (2010). Dynamic fuzzy neural networks modeling and adaptive backstepping tracking control of uncertain chaotic systems. *Neurocomputing*, *73*(16), 2873–2881. https://doi.org/https://doi.org/10.1016/j.neucom.2010.08.008

Lin, J., & Liao, G.-T. (2016). A modularized cable-suspended robot: Implementation and oscillation suppression control. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, *230*(9), 1030–1043. https://doi.org/10.1177/0959651816662157

Liu, W., Lin, Z., Lin, W., & Zheng, D. (2018). Demography and Human Ecology: Research on Regional Difference in Aging of Population in China. *Ekoloji*, *27*(106), 1323–1326.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11–26. https://doi.org/https://doi.org/10.1016/j.neucom.2016.12.038

Lo, H. S., & Xie, S. Q. (2012). Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects. *Medical Engineering & Physics*, *34*(3), 261–268.

Loureiro, R. C. V, Harwin, W. S., Nagai, K., & Johnson, M. (2011). Advances in upper limb stroke rehabilitation: a technology push. *Medical & Biological Engineering & Computing*, *49*(10), 1103.

Lowrey, K., Kolev, S., Dao, J., Rajeswaran, A., & Todorov, E. (2018). Reinforcement learning for non-prehensile manipulation: Transfer from simulation to physical system. In *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)* (pp. 35–42). IEEE.

Lum, P. S., Godfrey, S. B., Brokaw, E. B., Holley, R. J., & Nichols, D. (2012). Robotic approaches for rehabilitation of hand function after stroke. *American Journal of Physical Medicine & Rehabilitation*, *91*(11), S242–S254.

Ma, O., & Diao, X. (2005). Dynamics analysis of a cable-driven parallel manipulator for hardware-in-the-loop dynamic simulation. In *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on* (pp. 837–842). IEEE.

Maciejasz, P., Eschweiler, J., Gerlach-Hahn, K., Jansen-Troy, A., & Leonhardt, S. (2014). A survey on robotic devices for upper limb rehabilitation. *Journal of Neuroengineering and Rehabilitation*, *11*(1), 3.

Mao, Y., & Agrawal, S. K. (2010). Wearable Cable-driven Upper Arm Exoskeleton - Motion with Transmitted Joint Force and Moment Minimization, 4334–4339.

Mao, Y., & Agrawal, S. K. (2012). Design of a cable-driven arm exoskeleton (CAREX) for neural rehabilitation. *IEEE Transactions on Robotics*, *28*(4), 922–931.

Marchal-Crespo, L., & Reinkensmeyer, D. J. (2009). Review of control strategies for robotic movement training after neurologic injury. *Journal of Neuroengineering and Rehabilitation*, *6*(1), 20.

Masiero, S., & Armani, M. (2011). Upper-limb robot-assisted therapy in rehabilitation of acute stroke patients: focused review and results of new randomized controlled trial. *Journal of Rehabilitation Research and Development*, *48*(4), 355.

Mayhew, D., Bachrach, B., Rymer, W. Z., & Beer, R. F. (2005). Development of the MACARM - a novel cable robot for upper limb neurorehabilitation. In *International Conference on Rehabilitation Robotics (ICORR)* (pp. 299–302). https://doi.org/10.1109/ICORR.2005.1501106

Merlet, J.-P. (2010). MARIONET, a family of modular wire-driven parallel robots. In *Advances in Robot Kinematics: Motion in Man and Machine* (pp. 53–61). Springer.

Meunier, G., Boulet, B., & Nahon, M. (2009). Control of an overactuated cable-driven parallel mechanism for a radio telescope application. *IEEE Transactions on Control Systems Technology*, *17*(5), 1043–1054. https://doi.org/10.1109/TCST.2008.2004812

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., … Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)* (pp. 1928–1937).

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *ArXiv Preprint ArXiv:1312.5602*.

Moffaert, K. Van, Drugan, M. M., & Nowé, A. (2013). Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)* (pp. 191–199). https://doi.org/10.1109/ADPRL.2013.6615007

Mohamaddan, S., & Komeda, T. (2010). Wire-driven mechanism for finger rehabilitation device. In *IEEE International Conference on Mechatronics and Automation* (pp. 1015–1018). https://doi.org/10.1109/ICMA.2010.5588077

Montgomery, W. H., & Levine, S. (2016). Guided policy search via approximate mirror descent. In *Advances in Neural Information Processing Systems* (pp. 4008–4016).

Morales, R., Badesa, F. J., García-Aracil, N., Sabater, J. M., & Pérez-Vidal, C. (2011). Pneumatic robotic systems for upper limb rehabilitation. *Medical & Biological Engineering & Computing*, *49*(10), 1145.

Moreno, J. C., Brunetti, F., Navarro, E., Forner-Cordero, A., & Pons, J. L. (2009). Analysis of the human interaction with a wearable lower-limb exoskeleton. *Applied Bionics and Biomechanics*, *6*(2), 245–256.

Nagabandi, A, Kahn, G., Fearing, R. S., & Levine, S. (2018). Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 7559–7566). https://doi.org/10.1109/ICRA.2018.8463189

Nagabandi, Anusha, Yang, G., Asmar, T., Kahn, G., Levine, S., & Fearing, R. S. (2017). Neural Network Dynamics Models for Control of Under-actuated Legged Millirobots. *ArXiv Preprint ArXiv:1711.05253*.

Nagpal, Y.-L. P. and B. C. and N. O. P.-A. and D. Y. and L. S. and R. J. W. and E. C. G. and R. (2014). Design and control of a bio-inspired soft wearable robotic device for ankle–foot rehabilitation. *Bioinspiration & Biomimetics*, *9*(1), 16007.

Ng, A. Y., Harada, D., & Russell, S. (n.d.). Policy invariance under reward transformations: Theory and application to reward shaping.

Nguyen, D. Q., & Gouttefarde, M. (2014). Stiffness matrix of 6-dof cable-driven parallel robots and its homogenization. In *Advances in Robot Kinematics* (pp. 181–191). Springer. https://doi.org/https://doi.org/10.1007/978-3-319-06698-1_20

Niyetkaliyev, A. S., Hussain, S., Ghayesh, M. H., & Alici, G. (2017). Review on Design and Control Aspects of Robotic Shoulder Rehabilitation Orthoses. *IEEE Transactions on Human-Machine Systems*, *47*(6), 1134–1145. https://doi.org/10.1109/THMS.2017.2700634

Norouzi-Gheidari, N., Archambault, P. S., & Fung, J. (2012). Effects of robot-assisted therapy on stroke rehabilitation in upper limbs: systematic review and meta-analysis of the literature. *Journal of Rehabilitation Research and Development*, *49*(4), 479.

Oh, S.-R., & Agrawal, S. K. (2005). Cable suspended planar robots with redundant cables: Controllers with positive tensions. *IEEE Transactions on Robotics*, *21*(3), 457–465. https://doi.org/https://doi.org/10.1109/TRO.2004.838029

Osogami, T. (2012). Robustness and risk-sensitivity in Markov decision processes. In *Advances in Neural Information Processing Systems* (pp. 233–241).

Otis, M. J.-D., Perreault, S., Nguyen-Dang, T., Lambert, P., Gouttefarde, M., Laurendeau, D., & Gosselin, C. (2009). Determination and Management of Cable Interferences Between Two 6-DOF Foot Platforms in a Cable-Driven Locomotion Interface. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *39*(3), 528–544. https://doi.org/10.1109/TSMCA.2009.2013188

Ottaviano, E., & Castelli, G. (2010). A study on the effects of cable mass and elasticity in cable-based parallel manipulators. *ROMANSY 18 Robot Design, Dynamics and Control*, 149–156.

Park, S., Weber, L., Bishop, L., Stein, J., & Ciocarlie, M. (2018). Design and Development of Effective Transmission Mechanisms on a Tendon Driven Hand Orthosis for Stroke Patients. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2281–2287). https://doi.org/10.1109/ICRA.2018.8461069

Park, Y., Santos, J., Galloway, K. G., Goldfield, E. C., & Wood, R. J. (2014). A soft wearable robotic device for active knee motions using flat pneumatic artificial muscles. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4805–4810). https://doi.org/10.1109/ICRA.2014.6907562

Passalis, N., & Tefas, A. (2019). Deep reinforcement learning for controlling frontal person close-up shooting. *Neurocomputing*. Retrieved from https://doi.org/10.1016/j.neucom.2019.01.046

Pattanaik, A., Tang, Z., Liu, S., Bommannan, G., & Chowdhary, G. (2018). Robust Deep Reinforcement Learning with Adversarial Attacks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (pp. 2040–2042). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Pedrocchi, A., Ferrante, S., Ambrosini, E., Gandolla, M., Casellato, C., Schauer, T., … Gföhler, M. (2013). MUNDUS project: MUltimodal Neuroprosthesis for daily Upper limb Support. *Journal of Neuroengineering and Rehabilitation*, *10*(1), 66.

Peng, X. B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2018). Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1–8). https://doi.org/10.1109/ICRA.2018.8460528

Pennycott, A., Wyss, D., Vallery, H., Klamroth-Marganska, V., & Riener, R. (2012). Towards more effective robotic gait training for stroke rehabilitation: a review. *Journal of Neuroengineering and Rehabilitation*, *9*(1), 65.

Pham, C. B., Yeo, S. H., Yang, G., & Chen, I. M. (2009). Workspace analysis of fully restrained cable-driven manipulators. *Robotics and Autonomous Systems*, *57*(9), 901–912. https://doi.org/10.1016/j.robot.2009.06.004

Pinto, L., Davidson, J., Sukthankar, R., & Gupta, A. (2017). Robust Adversarial Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (pp. 2817–2826). JMLR.org.

Pons, J. L. (2010). Rehabilitation exoskeletal robotics. *IEEE Engineering in Medicine and Biology Magazine*, *29*(3), 57–63.

Prange, G. B., Jannink, M. J. A., Groothuis-Oudshoorn, C. G. M., Hermens, H. J., & Ijzerman, M. J. (2009). Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke. *Journal of Rehabilitation Research and Development*, *43*(2), 171–184.

Prieto, A., Prieto, B., Ortigosa, E. M., Ros, E., Pelayo, F., Ortega, J., & Rojas, I. (2016). Neural networks: An overview of early research, current frameworks and new challenges. *Neurocomputing*, *214*, 242–268. https://doi.org/https://doi.org/10.1016/j.neucom.2016.06.014

Proietti, T., Crocher, V., Roby-Brami, A., & Jarrassé, N. (2016). Upper-limb robotic exoskeletons for neurorehabilitation: a review on control strategies. *IEEE Reviews in Biomedical Engineering*, *9*, 4–14.

Pusey, J., Fattah, A., Agrawal, S., & Messina, E. (2004). Design and workspace analysis of a 6-6 cable-suspended parallel robot. *Mechanism and Machine Theory*, *39*(7), 761–778. https://doi.org/10.1016/j.mechmachtheory.2004.02.010

Rahman, M. D. M., Rashid, S. M. H., & Hossain, M. M. (2018). Implementation of Q learning and deep Q network for controlling a self balancing robot model. *Robotics and Biomimetics*, *5*(1), 8. Retrieved from https://doi.org/10.1186/s40638-018-0091-9

Randløv, J., & Alstrøm, P. (1998). Learning to Drive a Bicycle Using Reinforcement Learning and Shaping. In *International Conference on Machine Learning (ICML)* (Vol. 98, pp. 463–471).

Recht, B. (2018). A Tour of Reinforcement Learning: The View from Continuous Control. *ArXiv Preprint ArXiv:1806.09460*.

Recht, B., Re, C., Wright, S., & Niu, F. (2011). Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems* (pp. 693–701).

Riener, R. (2007). Robot-aided rehabilitation of neural function in the upper extremities. In *Operative Neuromodulation* (pp. 465–471). Springer.

Rosati, G., Secoli, R., Zanotto, D., Rossi, A., & Boschetti, G. (2008). Planar robotic systems for upper-limb post-stroke rehabilitation. In *ASME International Mechanical Engineering Congress and Exposition* (pp. 115–124). American Society of Mechanical Engineers.

Rosati, G., Zanotto, D., & Agrawal, S. K. (2011). On the design of adaptive cable-driven systems. *Journal of Mechanisms and Robotics*, *3*(2), 21004.

Salcudean, S. E., & Stocco, L. (2000). Isotropy and actuator optimization in haptic interface design. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* (Vol. 1, pp. 763–769). IEEE.

Salisbury, J. K., & Craig, J. J. (1982). Articulated Hands: Force Control and Kinematic Issues. *The International Journal of Robotics Research*, *1*(1), 4–17. https://doi.org/10.1177/027836498200100102

Sánchez, E., Díaz, I., & Gil, J. J. (2011). Lower-Limb Robotic Rehabilitation: Literature Review and Challenges. *Journal of Robotics*. https://doi.org/10.1155/2011/759764

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning (ICML)* (pp. 1889–1897).

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *ArXiv Preprint ArXiv:1707.06347*.

Shao, Z.-F., Tang, X., & Yi, W. (2014). Optimal design of a 3-DOF cable-driven upper arm exoskeleton. *Advances in Mechanical Engineering*, *6*, 157096.

Sharma, S. K., & Chandra, P. (2010). Constructive neural networks: A review. *International Journal of Engineering Science and Technology*, *2*(12), 7847–7855.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., … Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*, 484.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *International Conference on Machine Learning (ICML)* (pp. 387–395).

Tan, J., Zhang, T., Coumans, E., Iscen, A., Bai, Y., Hafner, D., … Vanhoucke, V. (2018). Sim-to-real: Learning agile locomotion for quadruped robots. *ArXiv Preprint ArXiv:1804.10332*.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 23–30). https://doi.org/10.1109/IROS.2017.8202133

Todorov, E., Erez, T., & Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5026–5033). IEEE. https://doi.org/https://doi.org/10.1109/IROS.2012.6386109

Tsagarakis, N. G., & Caldwell, D. G. (2003). Development and control of a 'soft-actuated'exoskeleton for use in physiotherapy and training. *Autonomous Robots*, *15*(1), 21–33.

Tsurumine, Y., Cui, Y., Uchibe, E., & Matsubara, T. (2019). Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation. *Robotics and Autonomous Systems*, *112*, 72–83. https://doi.org/https://doi.org/10.1016/j.robot.2018.11.004

UNFPA. (2012). *Ageing in the Twenty-First Century: A Celebration and A Challenge*. UNFPA.

Veneman, J. F., Ekkelenkamp, R., Kruidhof, R., van der Helm, F. C. T., & van der Kooij, H. (2006). A series elastic-and bowden-cable-based actuation system for use as torque actuator in exoskeleton-type robots. *The International Journal of Robotics Research*, *25*(3), 261–281.

Verhoeven, R., & Hiller, M. (2000). Estimating the Controllable Workspace of Tendon-Based Stewart Platforms BT - Advances in Robot Kinematics. In J. Lenarčič & M. M. Stanišić (Eds.) (pp. 277–284). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-011-4120-8_29

Vitiello, N., Lenzi, T., Roccella, S., De Rossi, S. M. M., Cattin, E., Giovacchini, F., … Carrozza, M. C. (2013). NEUROExos: A powered elbow exoskeleton for physical rehabilitation. *IEEE Transactions on Robotics*, *29*(1), 220–235.

Wang, T., Tong, S., Yi, J., & Li, H. (2015). Adaptive Inverse Control of Cable-Driven Parallel System Based on Type-2 Fuzzy Logic Systems. *IEEE Transactions on Fuzzy Systems*, *23*(5), 1803–1816. https://doi.org/10.1109/TFUZZ.2014.2379284

Wehner, M., Quinlivan, B., Aubin, P. M., Martinez-Villalpando, E., Baumann, M., Stirling, L., … Walsh, C. (2013). A lightweight soft exosuit for gait assistance. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3362–3369). https://doi.org/10.1109/ICRA.2013.6631046

Wei, W., Qu, Z., Wang, W., Zhang, P., & Hao, F. (2018). Design on the Bowden Cable-Driven Upper Limb Soft Exoskeleton. *Applied Bionics and Biomechanics*. https://doi.org/10.1155/2018/1925694

Williams, R. L., Gallina, P., & Vadia, J. (2003). Planar translational cable-direct-driven robots. *Journal of Field Robotics*, *20*(3), 107–120.

Xiao, L., & Liao, B. (2016). A convergence-accelerated Zhang neural network and its solution application to Lyapunov equation. *Neurocomputing*, *193*, 213–218. https://doi.org/https://doi.org/10.1016/j.neucom.2016.02.021

Xie, Z., Berseth, G., Clary, P., Hurst, J., & Panne, M. van de. (2018). Feedback Control For Cassie With Deep Reinforcement Learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1241–1246).

Xiong, H., & Diao, X. (2017a). Cable tension control of cable-driven parallel manipulators with position-controlling actuators. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 1763–1768). https://doi.org/10.1109/ROBIO.2017.8324673

Xiong, H., & Diao, X. (2017b). The effect of cable tensions on the stiffness of cable-driven parallel manipulators. In *IEEE/ASME international conference on advanced intelligent mechatronics* (pp. 1185–1190). IEEE. https://doi.org/https://doi.org/10.1109/AIM.2017.8014179

Xiong, H., & Diao, X. (2018a). Geometric isotropy indices for workspace analysis of parallel manipulators. *Mechanism and Machine Theory*, *128*, 648–662. https://doi.org/https://doi.org/10.1016/j.mechmachtheory.2018.05.017

Xiong, H., & Diao, X. (2018b). Motion Control of Cable-Driven Rehabilitation Devices with Large Deformation Cables. In *IEEE International Conference on Cyborg and Bionic Systems (CBS)* (pp. 537–543). https://doi.org/10.1109/CBS.2018.8612272

Xiong, H., & Diao, X. (2019). A review of cable-driven rehabilitation devices. *Disability and Rehabilitation: Assistive Technology*, 1–13. https://doi.org/10.1080/17483107.2019.1629110

Xiong, H., Ma, T., Zhang, L., & Diao, X. (2019). Comparison of End-to-End and Hybrid Deep Reinforcement Learning Strategies for Controlling Cable-Driven Parallel Robots. *Neurocomputing*.

Xiong, H., Zhang, L., & Diao, X. (2019). A Novel Control Strategy for Cable-Driven Parallel Robots with Unknown Jacobians. In *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*.

Xiong, H., Zhang, L., Liu, Z., & Diao, X. (2018). Joint Force Analysis and Moment Efficiency Index of Cable-Driven Rehabilitation Devices. In *IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* (pp. 1–5). https://doi.org/10.1109/HUMANOIDS.2018.8625029

Yan, T., Cempini, M., Oddo, C. M., & Vitiello, N. (2015). Review of assistive strategies in powered lower-limb orthoses and exoskeletons. *Robotics and Autonomous Systems*, *64*, 120–136. https://doi.org/10.1016/j.robot.2014.09.032

Yang, Y., & Yan, Y. (2015). Trajectory tracking for robotic airships using sliding mode control based on neural network approximation and fuzzy gain scheduling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, *230*(2), 184–196. https://doi.org/10.1177/0959651815617884

Yedavalli, R. (1985). Improved measures of stability robustness for linear state space models. *IEEE Transactions on Automatic Control*, *30*(6), 577–579. https://doi.org/10.1109/TAC.1985.1103996

Yegnanarayana, B. (2009). *Artificial neural networks*. PHI Learning Pvt. Ltd.

Yeow, C.-H., Baisch, A. T., Talbot, S. G., & Walsh, C. J. (2014). Cable-Driven Finger Exercise Device With Extension Return Springs for Recreating Standard Therapy Exercises. *Journal of Medical Devices*, *8*(1), 14502.

Yoon, J. W., Ryu, J., & Hwang, Y.-K. (2010). Optimum design of 6-DOF parallel manipulator with translational/rotational workspaces for haptic device application. *Journal of Mechanical Science and Technology*, *24*(5), 1151–1162. https://doi.org/10.1007/s12206-010-0321-8

Zanganeh, K. E., & Angeles, J. (1997). Kinematic isotropy and the optimum design of parallel manipulators. *International Journal of Robotics Research*, *16*(2), 185–197.

Zhang, B., Shang, W., Cong, S., & Liu, Y. (2017). Size optimization of the moving platform for cable-driven parallel manipulators based on stiffness characteristics. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *232*(11), 2057–2066. https://doi.org/10.1177/0954406217714819

Zhang, M., Geng, X., Bruce, J., Caluwaerts, K., Vespignani, M., SunSpiral, V., … Levine, S. (2017). Deep reinforcement learning for tensegrity robot locomotion. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 634–641). IEEE. https://doi.org/10.1109/ICRA.2017.7989079

Zhang, T., Kahn, G., Levine, S., & Abbeel, P. (2016). Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 528–535). IEEE. https://doi.org/10.1109/ICRA.2016.7487175

Zi, B., Ding, H., Cao, J., Zhu, Z., & Kecskeméthy, A. (2014). Integrated Mechanism Design and Control for Completely Restrained Hybrid-Driven Based Cable Parallel Manipulators. *Journal of Intelligent & Robotic Systems*, *74*(3), 643–661. https://doi.org/10.1007/s10846-013-9848-0

Zi, B., Duan, B. Y., Du, J. L., & Bao, H. (2008). Dynamic modeling and active control of a cable-suspended parallel robot. *Mechatronics*, *18*(1), 1–12.

PUBLICATIONS

Journal publications

Xiong, H, & Diao, X. (2019a). A review of cable-driven rehabilitation devices. **Disability and Rehabilitation: Assistive Technology**, 1–13. https://doi.org/10.1080/17483107.2019.1629110

Xiong, H, & Diao, X. (2019d). The effects of cables' strain and specific stiffness on the stiffness of cable-driven parallel manipulators. Proceedings of the Institution of Mechanical Engineers, Part C: **Journal of Mechanical Engineering Science**. https://doi.org/10.1177/0954406219846536

Xiong, H, Hu, J., & Diao, X. (2019b). Optimize Energy Efficiency of Quadrotors Via Arm Rotation. **Journal of Dynamic Systems, Measurement, and Control**, 141(9). https://doi.org/10.1115/1.4043227

Xiong, H, Ma, T., Zhang, L., & Diao, X. (2019). Comparison of End-to-End and Hybrid Deep Reinforcement Learning Strategies for Controlling Cable-Driven Parallel Robots. **Neurocomputing**. https://doi.org/10.1016/j.neucom.2019.10.020

Xiong, H, & Diao, X. (2018). Geometric isotropy indices for workspace analysis of parallel manipulators. **Mechanism and Machine Theory**, 128, 648–662. https://doi.org/https://doi.org/10.1016/j.mechmachtheory.2018.05.017

Xiong, H, & Diao, X. (2019b). Policy Safety Robustness of Reinforcement Learning: A View from Robust Control. **Neurocomputing**. (in review)

Xiong, H, & Diao, X. (2019c). Stiffness Analysis of Cable-Driven Parallel Mechanisms with Cables Having Large Sustainable Strains. Proceedings of the Institution of Mechanical Engineers, Part C: **Journal of Mechanical Engineering Science**. (in review)

Xiong, H, Hu, J., & Diao, X. (2019a). A Flight-Trim System for Quadrotors with Rotatable Arms. **Journal of Guidance, Control, and Dynamics**. (in review)

Xiong, H, Zhang, L., & Diao, X. (2019). A Learning-Based Control Framework for Cable-Driven Parallel Robots with Unknown Jacobians. Proceedings of the Institution of Mechanical Engineers, Part I: **Journal of Systems and Control Engineering**. (in review)

Conference publications

Xiong, H, Zhang, L., & Diao, X. (2019). A Novel Control Strategy for Cable-Driven Parallel Robots with Unknown Jacobians. In IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO).

Xiong, H, Zhang, L., Liu, Z., & Diao, X. (2018). Joint Force Analysis and Moment Efficiency Index of Cable-Driven Rehabilitation Devices. In IEEE-RAS International Conference on Humanoid Robots (Humanoids) (pp. 1–5). https://doi.org/10.1109/HUMANOIDS.2018.8625029

Xiong, H, & Diao, X. (2017a). Cable tension control of cable-driven parallel manipulators with position-controlling actuators. In IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 1763–1768). https://doi.org/10.1109/ROBIO.2017.8324673

Xiong, H, & Diao, X. (2017b). Geometric isotropy indices for workspace analysis of spatial parallel manipulators. In IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 1985–1990). https://doi.org/10.1109/ROBIO.2017.8324710

Xiong, H, & Diao, X. (2018a). Motion Control of Cable-Driven Rehabilitation Devices with Large Deformation Cables. In IEEE International Conference on Cyborg and Bionic Systems (CBS) (pp. 537–543). https://doi.org/10.1109/CBS.2018.8612272

Xiong, H, & Diao, X. (2018b). Safe Tension Control of Cable-Driven Rehabilitation Devices with Elastic Cables. In IEEE International Conference on Cyborg and Bionic Systems (CBS) (pp. 544–548). https://doi.org/10.1109/CBS.2018.8612208

Xiong, H, & Diao, X. (2017a). Novel geometric isotropy indices for workspace analysis of planar parallel manipulators. In IEEE International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 1191–1196). IEEE. https://doi.org/https://doi.org/10.1109/AIM.2017.8014180

Xiong, H, & Diao, X. (2017b). The effect of cable tensions on the stiffness of cable-driven parallel manipulators. In IEEE International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 1185–1190). IEEE. https://doi.org/https://doi.org/10.1109/AIM.2017.8014179

Li, S., **Xiong, H**., & Diao, X. (2019). Pre-Impact Fall Detection Using 3D Convolutional Neural Network. In IEEE International Conference on Rehabilitation Robotics (ICORR) (pp. 1173–1178). https://doi.org/10.1109/ICORR.2019.8779504

Ma, T., **Xiong, H**., Zhang, L., & Diao, X. (2019). Control of a Cable-Driven Parallel Robot via Deep Reinforcement Learning. In IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO). IEEE.

Zhang, L., Li, S., **Xiong, H**., Diao, X., Ma, O., & Wang, Z. (2019). Prediction of Intention Behind a Single Human Action: An Application of Convolutional Neural Network. In IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER).

Patents

Diao, X., & **Xiong, H**. (2019). Method and apparatus for cable-driven adaptive vibration control. US20190047705A1. (filed).