# ITERATIVE LEARNING CONTROL FOR LOWER LIMB EXOSKELETON ROBOT

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2022

**Zhongyi Wang**
Department of Electrical and Electronic Engineering

# Contents

# List of Tables

# List of Figures

# Notations

| | |
|---|---|
| $\epsilon$ | Coupling Strength Parameter |
| $\gamma$ | Learning Gain Matrix |
| $\lambda$ | Error Convergence Rate |
| $\mu$ | Oscillation Amplitude Parameter |
| $\omega$ | Oscillation Frequency |
| $\phi$ | Costate System Status Variable |
| $\tau$ | Output Torque of Actuators |
| $\theta$ | Angle Vector |
| $\xi$ | Feed-forward Item |
| $a_1, a_2$ | Coefficients in Feedback Linearisation |
| $c_1, c_2$ | Coefficients of Shank Part Dynamic Model |
| $C_r$ | Effects of Coriolis Force |
| $E_i$ | Remaining Error of Coupled Oscillators |
| $exp(\cdot)$ | Exponential Function |

|   |   |
|---|---|
| $f_0, b_0, h_0$ | Lipschitz constants |
| $G$ | Toeplitz Matrix |
| $G_i$ | Gravity Term |
| $K_d$ | Learning Coefficient |
| $K_l$ | Control Gain |
| $K_p$ | Proportional Learning Coefficient |
| $L_l$ | Observer Gain |
| $m_f$ | Mass of Foot Part |
| $M_r$ | Inertial Matrix |
| $m_s$ | Mass of Shank Part |
| $O_i$ | Individual Oscillator in Coupled Oscillators |
| $T$ | Average Gait Cycle Time |

# Abbreviations

**ALEX** Advanced Lower Extremity Exoskeleton.

**AO** Adaptive Oscillator.

**BLEEX** Berkeley Lower Extremity Exoskeleton.

**CEF** Composite Energy Function.

**CoM** Centre of Mass.

**CPG** Central Pattern Generator.

**D-Type** Derivative-Type.

**DIMS** Dual Internal Model Structure.

**DoF** Degree of Freedom.

**EMG** Electromyography.

**FL** Feedback Linearization.

**GLC** Global Lipschitz Continuous.

**HAL** Hybrid Assistive Limb.

**ILC** Iterative Learning Control.

**LLC** Local Lipschitz Continuous.

**LQR** Linear Quadratic Regulator.

**MIMO** Multi-Input Multi-Output.

**NOILC** Norm Optimal Iterative Learning Control.

**P-Type** Proportional-Type.

**PD-Type** Proportional-Derivative-Type.

**PI-Type** Proportional-Integral-Type.

**PID-Type** Proportional–Integral–Derivative-Type.

**PMSM** Permanent-Magnet Synchronous Machine.

**RoM** Range of Motion.

**SISO** Single-Input Single-Output.

**UAV** Unmanned Aerial Vehicle.

# Abstract

This thesis mainly focuses on the motion control problems of the wearable lower limb exoskeleton robot. Firstly, by approximating the ideal human gait as a cyclic signal, the motion control problem of the exoskeleton robot is converted to a tracking problem in a series of fixed and finite time intervals, where each interval represents a gait cycle. Then, a novel iterative learning control algorithm has been proposed. The proposed algorithm combines the concept of feedback linearization with the iterative learning control method and has advantages in response speed, tracking accuracy, and error convergence rate. Relative simulation results are given to demonstrate the effectiveness of the proposed algorithm. Finally, the adaptive oscillator method which is also feasible for the problem has been introduced and a comparison between the two methods is given. Both advantages and disadvantages of the proposed iterative learning control algorithm are discussed.

# Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright Statement

# List of Publications

**1).** **Zhongyi Wang**, and Zhengtao Ding. ”Iterative Learning Control for the Shank Part of Lower Limb Exoskeleton.” *2019 Chinese Control Conference (CCC)*, pp. 668-673. IEEE, 2019.

# Acknowledgements

First and foremost, I would like to thank Professor Zhengtao Ding for his supervision in these years. He led me through the door of control theory study when I was still an MSc student. In the next few years, he pointed out the new research field to form my PhD project. Besides, he is also a very kind supervisor who really cares about my feelings and supported me through my most depressing times.

I would also like to thank Yi Dong, Erbao Lu, Yuyang Zhou, Yiqiao Xu, and all my colleagues and friends at the University of Manchester, for their help and kind encouragement in all those years.

My thanks also go to Haowei Zhang, Rong Zhou, Zijiang Yang, Xiaoyao Song, and Jie Ma. They all have been my friends for many years. Their help and support also warmed me so much in my darkest times.

Finally, I would like to thank my parents. I would have not gone so far and completed this project without their encouragement and financial support.

To My Parents

# Chapter 1

# Introduction

## 1.1 Background

The desire to be physically powerful has a long history for human beings. In ancient times, having a stronger body usually means higher efficiency in hunting, gathering, or fishing. Later, when human beings become more civilized, a stronger body may imply being more productive in farming or working and a better chance to prevail over others in battle, all of which will contribute to a higher possibility of survival [1]. Hence, having a more sturdy, more muscular physique has been a target of people for thousands of years, even throughout the entire history of the human race. In the past, efforts to improve body strength generally include doing physical exercises and sport, as well as having a more balanced diet. Though these methods can be quite effective, there always exists an upper bound for their final results, which is, in fact, the physiological limitations of human bodies. For example, the condition of the human respiratory system will influence the endurance exercise performance of the human body [2], and the bones, muscles, and ligaments will influence the safety load of human limbs [3]. Those are the limitations that can not be overcome by the human body itself. People have to think of using new machines and new technologies to break those barriers. In modern times, many have laid their eyes down on robots, especially the exoskeleton robot.

The concept of building an exoskeleton robot has been growing for more than one hundred years [4]. In the early stage of its development, the progress is minor. It was only until the 1960s after the modern robots have been transferred from science fiction to reality [5], that the research of exoskeletons had been sped up. During the last two decades, the rapid development of mechanic, electrical and electronic technologies has accelerated the evolution of exoskeleton robots on a massive scale. A great rise has been witnessed in the number of functional exoskeleton robots. They have been used not only as a common tool in manufacturing industries but also as helpful assistants in health care, rescue service, heavy load carrying, rehabilitation, and many other aspects [6]. The flourishing research of relative control problems has also improved the performance of the new exoskeleton robots, benefiting them to better enhance the capability of human bodies and overcome their physical limitations, which has led to a wider market of those exoskeleton robots as well.

The exoskeleton robots are defined as a group of wearable devices that can be attached to human bodies as extra skeleton systems in order to improve the human body performance in physical tasks [7]. Some pioneer studies of exoskeleton robots were also started in the late 1960s [8]. However, those more practical and reliable cases were mainly started in the last two decades, including some prototype exoskeleton robots built by the Israels [9], the Hybrid Assistive Limb (HAL) robot by the Japanese [10], the Advanced Lower Extremity Exoskeleton (ALEX) robot by the University of Melbourne in Australia [11], and the Berkeley Lower Extremity Exoskeleton (BLEEX) project carried out by the University of California, Berkeley, America [12]. Aside from these, many other companies and institutions have begun their researches on exoskeleton robots in recent years. These studies include exoskeleton robots for both upper and lower limbs and cover a wide range of real-life application scenarios, such as rehabilitation, health care, walking assistance, and human power enhancement [13, 14].

Since the wearable exoskeleton robots have achieved great outcomes in providing assistance to human movements and augmentation of physical endurance, it is predictable that more relative researches will be carried out in future years. Further development with new types and models will also be continued. Hence, the study of relative control problems in exoskeleton robots will be of great importance as well.

## 1.2 Motivation

Exoskeletons are not equal to exoskeleton robots. In fact, exoskeletons can be categorized in several ways [15]. According to the presence of actively powered actuators or not, they are classified as active powered exoskeletons and passive exoskeletons. Usually only the former will be regarded as exoskeleton robots, which is also the research focus of this thesis as well.

In general, the passive exoskeletons use springs and dampers or other mechanical actuation to store and restore energy in order to drive the exoskeleton system [16]. Since the passive exoskeletons do not require any external power source, they take advantages in terms of simplicity, reliability, low cost, and much longer working endurance [17–19]. However, compared to the actively powered exoskeletons, the passive exoskeletons have reduced function and performance capabilities, providing a lower level of assistance than the active exoskeletons. Besides that, the passive exoskeletons need their users to execute the movement and provide a portion of the driving force to work. Hence they are less suitable in some circumstances, such as rehabilitation, where the users may lack sufficient strength to perform the activities required. In this case, despite the fact that some of the burdens have been relieved, they may still find the remaining part of the driving force exhausting or tiring enough.

On the contrary, actively powered exoskeletons, specifically the exoskeleton robots, are more capable of providing firm assistance for human movements. The actuators mounted on the exoskeleton robots can generate proper assistant torques depending on the user posture and external load. By adjusting the control strategy, exoskeleton robots are more flexible in dealing with different task conditions. They have more potential in generating larger supporting forces and superior versatility as well [20]. Exoskeleton robots are also more advanced than human beings in precision and repeatability, which makes them suitable for physical rehabilitation applications and therefore reduces the work burden of relative staff [11]. In industrial applications, exoskeleton robots have reduced the physical load on human limbs for a wide range of occupational activities and various postures [21]. In the meantime, extensive researches and applications of exoskeleton robots have also promoted the study of relative control issues and proposed many special requirements.

The actual design of an exoskeleton robot varies with its purpose, relative body part, structure, and the power technology applied. It can be used for rehabilitation, protection, enhancement, or assistance purpose. It can cover only one hand, one leg, the entire upper or lower limbs, or even the whole body. Its structure can be hard-body or soft suit, while the power source of its actuators can be hydraulic, pneumatic, electrical, or mechanical [15]. The numerous variations of exoskeleton robots have raised different requirements for the control design of each specific exoskeleton type. But they still share many things in common.

On the one hand, the structure of hard-body exoskeleton robots, one of the focuses of this thesis, has a certain level of similarities to the normal humanoid robots or robotic manipulators. They both usually consist of a series of links connected by revolving joints [22], while the joints are driven by powered actuators. They both need the angle sensors and accelerometers mounted on the body of robots to provide relative motion data, then to make the corresponding links to trace certain trajectories. Hence, the control designs of exoskeleton robots can be similar to normal robotic manipulators in some aspects, including dynamic model building, joint actuator control, motion control, and trajectory following.

On the other hand, there are numerous unique problems in the control of exoskeleton robots. One major difference between exoskeleton robots and traditional robotic manipulators is that exoskeletons are attached to human limbs, which as a result, brings an extra requirement in human-robot interaction. Instead of interfering with each other, human limbs and the exoskeleton robots should have a high degree of collaboration. Following that, one of the main motivations for investigating the control problems of exoskeleton robots is to achieve better coordination between the exoskeleton robot and human limbs [23, 24].

Therefore, the objective of the control issues of exoskeleton robots is to perform a faster, more accurate response to the designated trajectory, while the movement of the exoskeleton should coordinate with human limbs to make the user feel more comfortable. As the iterative learning based algorithm can adapt to the variation in references, fit to the different parameters in the model, and achieve high control accuracy after a few iterations, it is suitable to be applied in such scenarios.

Meanwhile, the control scheme should not be too complicated in order to achieve a higher speed of operation and better performance. The computational capability needed for the scheme should be low to reduce the cost of the exoskeleton-board processor so that the exoskeleton can be more affordable. The iterative learning based algorithm has much lower computational requirements than other learning algorithms while it can still do online learning and have good control performance. All those things have inspired the research of this thesis.

## 1.3 Contributions

The research investigates the motion control problem of the lower limb exoskeleton robots. The main contributions of this thesis are summarised as follows:

- A novel iterative learning based control algorithm has been proposed to deal with the motion control problem of the lower limb exoskeleton robots. The proposed algorithm has advantages in response speed and error convergence rate compared to traditional iterative learning control algorithms.
- The proposed novel algorithm has introduced the feedback linearization method to deal with the nonlinearity that exists in the system model to achieve a better control performance. Such a method has improved the average error value, error convergence rate, and algorithm robustness as well.
- The proposed algorithm has been applied to a model of the lower limb exoskeleton robot with nonlinear dynamics. Compared to other linearization methods, the proposed algorithm is more effective while its calculation is not complicated. The nonlinearities in the dynamic model will usually lead to a reduction of performance for the normal iterative learning algorithms. However, the proposed algorithm can handle those nonlinearities and perform a satisfactory result.
- An adaptive oscillator based control algorithm for the exoskeleton robot has also

been designed, which is another feasible method for this application. The proposed algorithm has demonstrated its advantages in tracking accuracy and error convergence rates. A comparison between the adaptive oscillator based method and the iterative learning based method has been carried out.

## 1.4 Thesis Outline

The organization of each chapter is described in detail at the beginning of those chapters. To understand the structure of the entire thesis, a general overview is presented in this section.

Chapter 1 Introduction. Brief overviews of the background and motivations for the development of the exoskeleton robot are shown in this chapter, as well as its motion control problems.

Chapter 2 Review and problem descriptions. The studies of the exoskeleton robot and its motion control problems over the past few decades have been reviewed in this chapter. The two main methods discussed in this thesis, the iterative learning control method and the adaptive oscillator method, have also been introduced.

Chapter 3 Preliminaries. The background preliminaries related to the iterative learning control method and the motion control problem of the exoskeleton robots are introduced in this chapter. A dynamic model for human lower limbs and exoskeleton limbs has been established. Finally, the acquisition of the human gait data which serves as the reference signal to the exoskeleton robot has also been presented.

Chapter 4 Iterative learning control for the shank part of the lower limb exoskeleton. A novel iterative learning algorithm is proposed to deal with the motion control problem of the lower limb exoskeleton robot. The proposed algorithm combines the feedback linearization principle with the original ILC algorithms and demonstrates an obvious advantage in error convergence rate. Simulation results are given to verify the effectiveness of the algorithm.

Chapter 5 Adaptive oscillator control for lower limb exoskeleton. Another feasible method for the problem, the adaptive oscillator method, is introduced in this chapter. A comparison between the adaptive oscillator method and the iterative learning method is carried out, as well as relative simulation results. The advantages and disadvantages of both methods are introduced.

Chapter 6 Conclusions and future works. This chapter concludes the contributions and the main results of this thesis. The possible remaining works and future research topics are discussed for further studies.

## 1.5 Conclusion

This chapter firstly introduces the background of exoskeleton robot studies as well as its motivations. Then, the main contributions of the thesis are summarised. Finally, the overall structure of the thesis is outlined by chapters.

# Chapter 2

# Review and Problem Descriptions

Over the last few decades, researches into the control issues of exoskeleton robots have expanded rapidly. This chapter briefly introduces the motion control problem of lower limb exoskeleton robots, as well as the two main control methods used in this thesis to address it. Section 2.1 is the introduction to the motion control problem. Section 2.2 provides a review of the two main control methods. Section 2.3 summarizes the chapter.

## 2.1 Motion Control Problem of the Lower Limb Exoskeleton Robot

Despite their flexibility in most scenarios in daily life, human bodies are still facing physical limitations in many cases. The strength of muscles and bones is strongly dependent on the age and health conditions of the relative human body. Sometimes they might also be overburdened by heavy loads. The demands of overcoming those physical limitations have motivated the development of wearable exoskeleton robots. By imitating the human skeleton with an additional exoskeleton robot system, the exoskeleton can reduce the load and energy consumption of its wearer during the movement, extend the walking distance and duration of the elder, mitigate the burden of workers, and provide medical rehabilitation service for those who have physical disabilities [11,25–27]. In the recent decade, the studies of exoskeleton robots from both the hardware and control algorithm aspects have drawn the attention of many researchers. Some of the researchers have also built several prototypes of exoskeleton robots and performed a lot of tests on their designs, structures, sensors, power sources, and control schemes [6, 7, 28–31].

The normal components of a lower limb exoskeleton robot are demonstrated in Figure 2.1. It usually includes the assistant actuators on the hip joint and knee joints, supporting structures along the thigh part, shank part, and foot part of the human lower limb, the holding structure around hip position, the power unit drives the entire exoskeleton system, as well as the controller that controls the actuators and manages the power supply [32–36]. There is no actuator on the ankle positions because the ankle joints of the exoskeleton are usually not power-assisted. Other payloads are usually positioned on the back of the wearer and supported by the exoskeleton structure.

The control system design of the exoskeleton robot is of great importance. A general scheme of the control system is demonstrated in Figure 2.2. The main functions of the controller are as follows: drive and control the joint actuators, control the motion of the exoskeleton robot, monitoring the motion and mechanical status for both the wearer and the exoskeleton robot, manage the power supply, and communicate with the host computer if necessary. To accomplish these functions, the control system of the exoskeleton robot usually includes several modules: the motion control module, the motion measurement module, the joint actuator drive and control module, the power management module, and the communication module [34, 37–40].

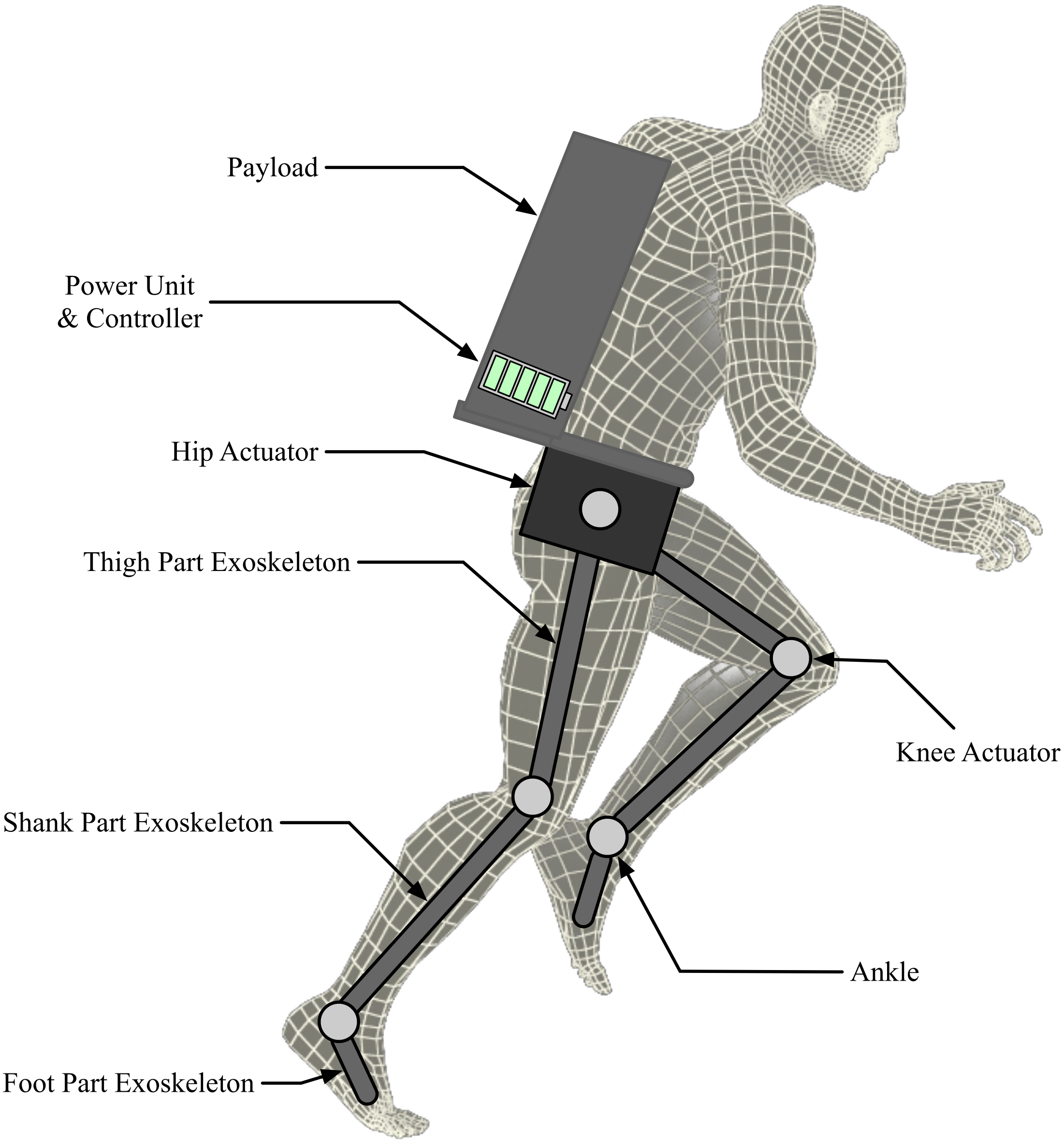

Figure 2.1: A Conceptual Sketch for the Components of A Lower Limb Exoskeleton.

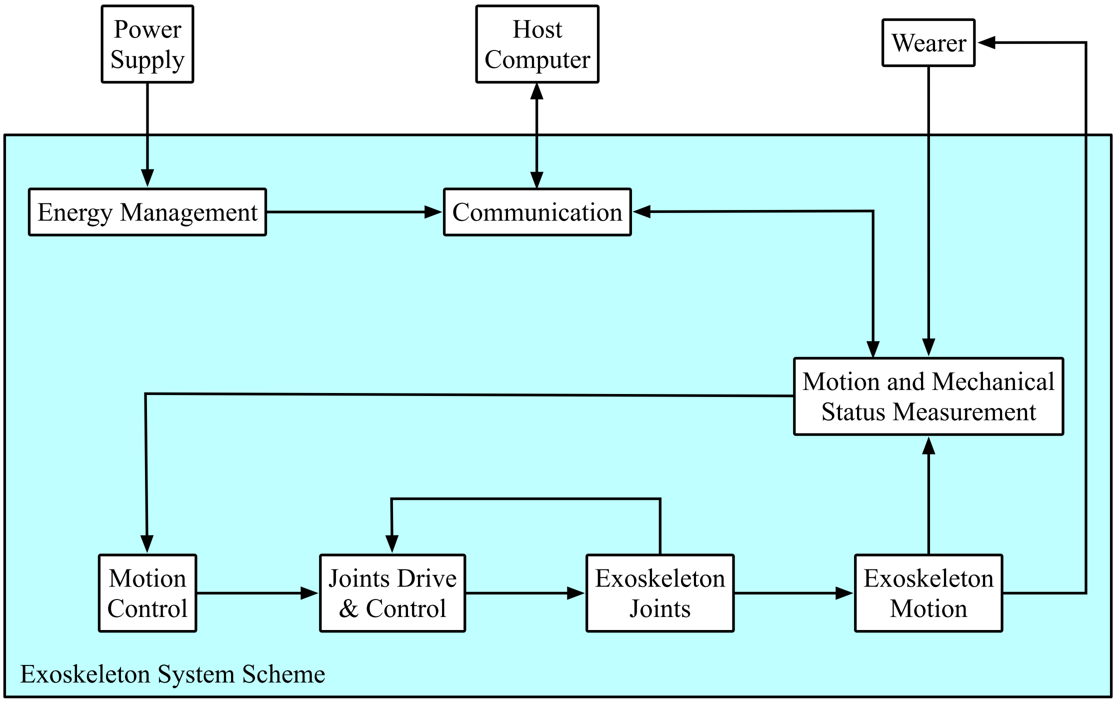

Figure 2.2: Control Scheme of the Exoskeleton Control System.

The motion control module and the joint actuator control module work as a cascade close-loop control system, where the joint actuator control module serves as the inner loop and the motion control module as the outer loop. The motion control module shall calculate proper trajectories for the exoskeleton limbs, provide a designated level of assistance, and make sure the movement of exoskeleton limbs can coordinate with the body of the wearer.

In the meantime, the motion measurement module shall detect the motion status for both the wearer and the exoskeleton limb, and provide these data as feedback signals to the motion control module. The power management module is responsible for providing enough power to drive all the control modules and actuators. Finally, the communication module can collect data from other modules and send them back to the host computer for storage and analysis. If necessary, it can also receive commands from the host computer and transmit those indications to the motion control module.

The kinds of actuators mounted on the exoskeleton vary according to the design of the exoskeleton. They can be hydraulic [41, 42], pneumatic [43, 44], electrical [45, 46], electroactive polymers [47, 48], etc. while electrical motors are the most common selection [15]. Whatever type the actuators are, their purpose must drive the joints and limbs of the exoskeleton while following the trajectory proposed in the motion control part.

The motion control module of the exoskeleton robot plays the most important role in deciding the effectiveness of the assistance and whether the wearer is comfortable. However, the control objectives are different for exoskeleton robots depending on their purposes and the relative body segments they are covering. For the upper limb exoskeleton robots, their main application fields are rehabilitation, power assist, human strength amplification, and haptic interactions [49–51]. Except for the rehabilitation cases, the main control objectives for most upper limb application cases are to identify human motion intention correctly and to perform proper assistance based on such intention. The motions of human upper limbs are usually not highly repetitive. In the meantime, the complexity of the human upper limb anatomic structure and the flexibility of upper limb motions make the problem even more complicated [52].

But for lower limb exoskeleton robots, the conditions for the control algorithm design are different. Rehabilitation has become the major application field of lower limb exoskeleton robots, where the motions of exoskeleton robots are usually periodic in these cases [53–55]. Besides, the human walking gait itself is very close to a periodic signal, though the characteristics are distinct for each individual [56, 57]. Hence even for the cases that the lower limb exoskeleton robots are not designed for rehabilitation purposes, for example, in walking endurance enhancement cases and heavy load carry cases, the motions of both human lower limbs and the exoskeleton robots are still highly repetitive. Such a repetitive characteristic can benefit the control design of lower limb exoskeleton robots on a great scale. Also, the degree of freedom requirement in exoskeletons design for the lower limb is less than the requirement of upper limbs [58, 59], which can simplify the control design as well.

One of the key features of the motion control design of the lower limb exoskeleton robots is determining how the exoskeleton robots should cooperate with the human limbs. The control method is usually chosen based on the status of cooperation and the design purpose of the exoskeleton robots.

On the one hand, if the exoskeleton robots are supposed to assist the wearers who still have their walking abilities or to improve the performances of the wearers in specific conditions, the wearers' decision on motion will have a much higher weight in motion control of the exoskeleton robots. For such a mode, the flowchart of the exoskeleton control algorithm is shown in Figure 2.3. In a few cases, the exoskeleton robots have totally handed the motion control role to the wearers, as their motions are, in fact, fully driven by the wearers. The exoskeleton robot will support the payload, while their algorithms are simply designed to follow any movement of the wearers [60]. In other cases, The control signals can be derived from the wearer based on various measurement methods, including the joint torque signal of the wearer [61–63], the interactive force between the exoskeleton robot and the body of the wearer [64, 65], and the sensitivity function from the wearer to the exoskeleton robot [29]. Also, there are many others that are designed to be controlled by the biological signals of the wearer, including surface Electromyography (EMG) signals [66–68] and brain-machine interface [69–72].

On the other hand, if the exoskeleton robots are supposed to assist the wearers who have damaged or even already lost their walking abilities, for example, in rehabilitation cases [73], the exoskeleton robots selves will then play the decisive role in motion control. For such a mode, the flowchart of the exoskeleton control algorithm is shown in Figure 2.4 In these cases, their motion control algorithms are usually designed to follow predefined trajectories that are imitating normal human gaits. The amplitude, frequency, and stride length of the gait can be adjusted in order to fit the specific gait characteristics of the wearer. Then, the control protocol can be derived based on either the position tracking error [74] or the joint angle tracking error [75].

Naturally, when a human walks forward straight and steadily on even ground at a constant speed, the general walking status can be regarded as a periodic procedure [76–78], and the time length of each period is the time length of each gait. According to the coordinate system rigidly fixed on the human body, during such a procedure, the position changing procedure and the joint angle variations of the lower limbs can be approximated as periodical signals [79–81]. These periodic gait signals can serve as the reference signals in the motion control problem of the exoskeleton robots. In this way,

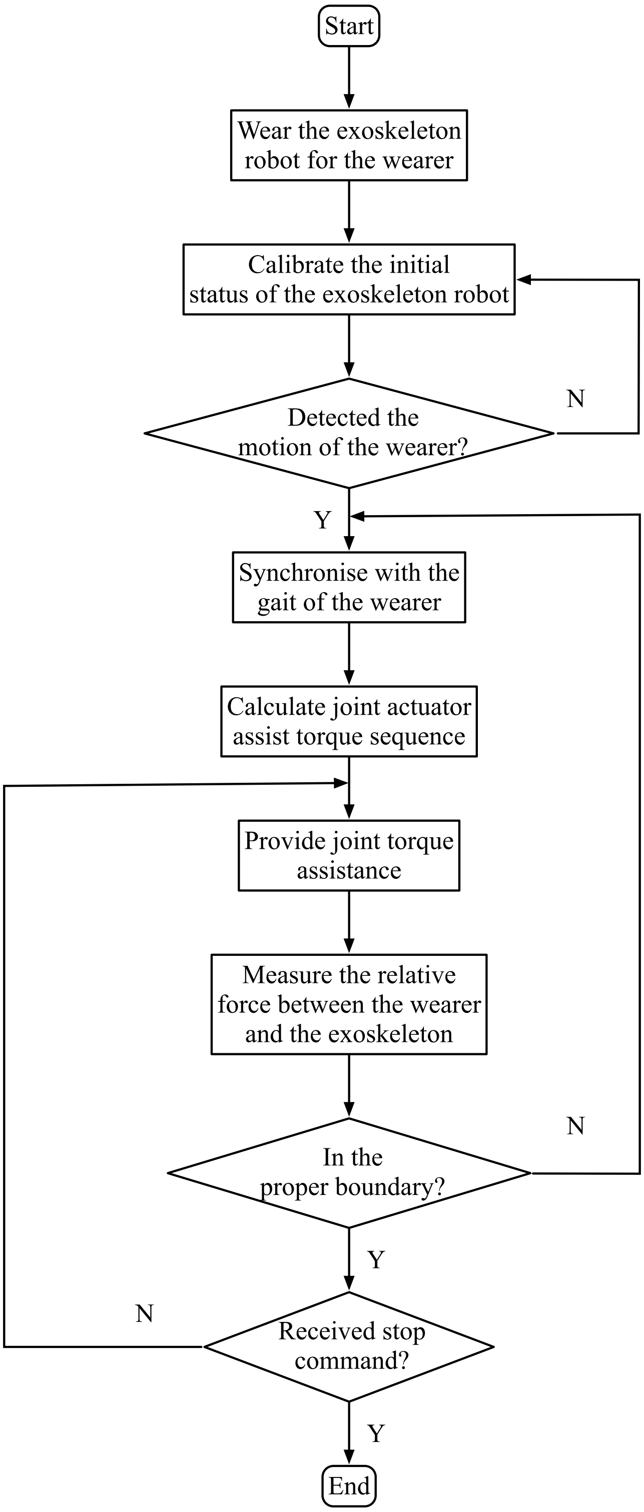
Start
Wear the exoskeleton robot for the wearer
Calibrate the initial status of the exoskeleton robot
Detected the motion of the wearer?
N
Y
Synchronise with the gait of the wearer
Calculate joint actuator assist torque sequence
Provide joint torque assistance
Measure the relative force between the wearer and the exoskeleton
N
In the proper boundary?
Y
N
Received stop command?
Y
End

Figure 2.3: Flowchart of the Control Algorithm in Assistant Cases.

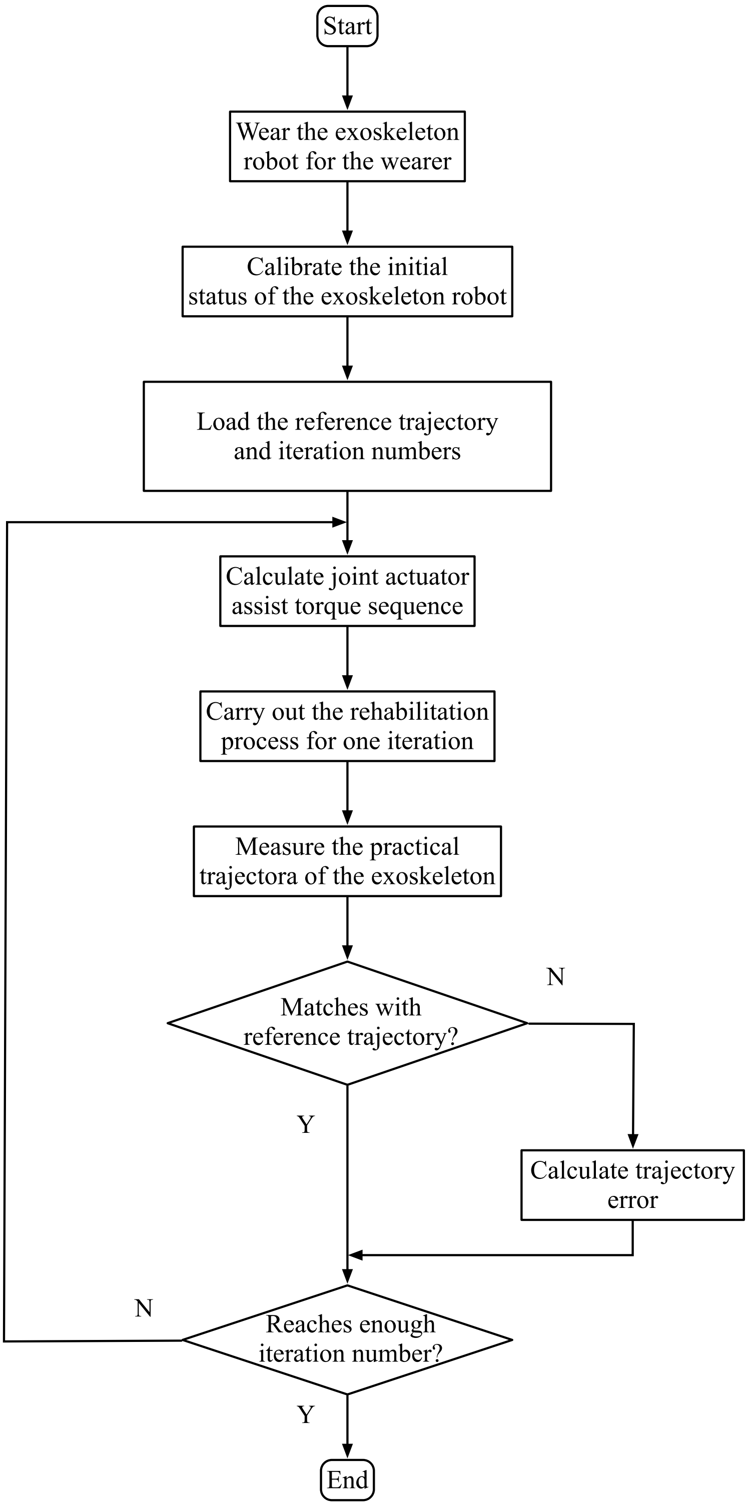
Start
Wear the exoskeleton robot for the wearer
Calibrate the initial status of the exoskeleton robot
Load the reference trajectory and iteration numbers
Calculate joint actuator assist torque sequence
Carry out the rehabilitation process for one iteration
Measure the practical trajectora of the exoskeleton
Matches with reference trajectory?
N
Y
Calculate trajectory error
N
Reaches enough iteration number?
Y
End

Figure 2.4: Flowchart of the Control Algorithm in Rehabilitation Cases.

the motion control problem of the lower limb exoskeleton robots can be transformed into a reference tracking problem in a fixed time interval. Therein, the time length of each interval is given by the period time of each gait cycle, and the reference trajectories will be the position data or joint angle data obtained from each period of the general periodical human gait.

However, since the development of practical and reliable exoskeleton robots was mainly started in just one or two decades, while these techniques also have not been widely applied in industrial and daily life scenarios, there is currently no international standard for the assessment of exoskeleton yet. In this thesis, the assessment features of the control algorithms include response speed, error convergence rate, final tracking accuracy, and robustness. The tracking error should be less than $5\%$ while such a response should be obtained in no more than 5 iterations. The final tracking error after 50 iterations should be less than $2\%$. In order to deal with such a tracking problem, two main methods are introduced in this thesis, the iterative learning control method and the adaptive oscillator method, respectively.

## 2.2 Applied Motion Control Methods

For the motion control problem of the lower limb exoskeleton robot proposed in the previous section, the key is to perform an ideal tracking for the exoskeleton robot to follow the target gait trajectory during each period. In the meantime, the calculation procedure of the motion control algorithm should not occupy too many computational resources from the robot-board microprocessor, such that the system cost can be reduced and the response can be sped up. Hence, two main control methods have been studied in this thesis to solve this problem.

### 2.2.1 Iterative Learning Control Method

The study of Iterative Learning Control (ILC) can be traced back to the early 1980s [82, 83]. The concept of ILC is that, for systems that have to complete the same task repetitively, the knowledge obtained from the previous control trials should be used in order to design a better control protocol for the next trial [84]. For example, an industrial manipulator has to execute an action repeatedly during a certain working period on an assembly line. The working period is long enough and can be divided into a series of fixed and finite time intervals, the length of which is the time required for the manipulator to accomplish the proposed action once. Then, with the help of the ILC method, the control performance of the manipulator can be improved iteratively by learning the control result from the previous time interval. As the number of iterations increases, the tracking error between the system response and reference will decrease.

ILC methods can be categorized in many ways [85, 86]. From the way of dealing with previous error data, or in other words, from the form of updating laws, there are many kinds of ILC algorithms including Proportional-Type (P-Type) [87], Derivative-Type (D-Type) [88], Proportional-Integral-Type (PI-Type) [89], Proportional-Derivative-Type (PD-Type) [90], and Proportional–Integral–Derivative-Type (PID-Type) [91], which has a similar categorization way like the general PID controllers. Combining the optimization methods with ILC, there are Norm Optimal Iterative Learning Control (NOILC) [92] and quadratic ILC [93]. Besides, there are also many studies covering different aspects of ILC design, including disturbance rejection [94], measurement noise [95], non-minimum phase [96], time delay [97], robustness [98], etc. Although the detailed implementation of ILC varies, they still share the same major feature that the performance of the system will be generally improved with the control errors obtained from previous control trials. Also, prior knowledge of the system model is not essential for the ILC method, though it can perform better if the model is known [99].

Such properties of the ILC methods soon lead to their applications in robotics, especially when the robot has to execute routine orders in a repetitive process. Typical application scenarios of the ILC methods include gantry robots [92], conveyor systems [100], Permanent-Magnet Synchronous Machine (PMSM) [101], and obviously the industrial manipulators [102–104] as well. The ILC method has presented a high level of tracking accuracy in those applications.

For the motion control problem of lower limb exoskeleton robots, the periodical characteristic of the human gait trajectory has made it suitable for the application of the ILC methods. Each step during walking can be regarded as a time interval, while the gait trajectory data of the wearer and the exoskeleton robot obtained from the previous step can be used in improving the performance of the exoskeleton robot during the next step. Compared to the traditional control methods, the ILC methods are more advanced in eliminating the phase lag and achieving a better tracking result. Also, if the gait pattern of the wearer changes during walking, the coefficients of the controller do not have to be readjusted since the ILC controller can generally adapt to the new gait pattern as the walking procedure continues. Furthermore, unlike the neural network based control methods, the ILC methods do not need any time-consuming training procedure in advance. The learning procedure of the ILC methods can be carried online. In addition, not only the computing time but also the computational resources needed for the ILC methods are much less than those of the neural network based methods, which makes it much easier for the ILC methods to be carried out by exoskeleton board processors. By reducing the requirement of the processors, the cost of the entire exoskeleton robot system can also be reduced.

Still, the common ILC methods also have their drawbacks. It can be hard for the ILC methods to obtain a stable control output if the dynamics of the system have strong nonlinearities. Meanwhile, if the learning parameters of the ILC algorithms have not been properly adjusted, sometimes the error convergence rates of the ILC methods can be very slow, or even goes unstable. To solve these problems, the novel ILC method is proposed in this thesis.

### 2.2.2 Adaptive Oscillator Method

The Adaptive Oscillator (AO) method is another way that fits for dealing with the motion control problem of the exoskeleton robots. The theory of adaptive oscillators has been studied by researchers since the 1990s [105]. By its features, an oscillator can be generally regarded as an autonomous dynamic system with one or more limit cycle attractors [106]. Then, adaptive oscillators are special forms of oscillators that have the ability to synchronize with other oscillators or external periodic signals [107, 108]. Compared to the normal oscillators, the adaptive oscillator has added extra dynamics to its parameters, which lead to an effective learning mechanism for the frequency. In this way, the frequency of the adaptive oscillator can generally adapt to the frequency of the input external periodic signal and maintain it even after the external periodic signal disappeared [109]. Due to their ability to learn external frequency signals, adaptive oscillators have been widely applied in the signal processing fields, including beat perception [110], pattern generator [111], noise filter [112], vibration compensation [113], and robotics [114, 115].

Oscillators are said to be coupled if they interact with each other. By combining a set of adaptive oscillators together, coupled adaptive oscillators can not only learn the frequencies of the external signals but also imitate their amplitudes and patterns [116], which has expanded their applications remarkably.

Because of these properties, the adaptive oscillator method is well suited for situations in which control systems must deal with a series of periodic signals. As stated in previous sections, the human gait signal is similar to a periodic signal for the motion control problem of lower limb exoskeleton robots. The relative control signals for joint actuators are periodical as well. The adaptive oscillators can learn the angle trajectories of human lower limbs during normal gaits and generate corresponding control signals for joint actuators. When the rhythm of gait changes, the adaptive oscillators can also learn the new characteristics and adapt to the new rhythm in a short time interval [117, 118].

Similar to the ILC method, the adaptive oscillator method also has advantages in that the learning process is online, no previous training is needed, and the exact system model is not essential [114, 119, 120]. Another important feature of the adaptive oscillator method is that it will not cause any phase lag during the control procedure [121, 122]. Meanwhile, the drawbacks of the adaptive oscillator are laid on response speed and final control accuracy. The adaptive oscillator controller usually will have remaining errors, which will become even more severe if the order of the adaptive oscillator is not high enough. Since both the ILC method and adaptive oscillator method can be applied to solve the control problem of the exoskeleton robot, their performances have also been compared in this thesis.

## 2.3 Conclusion

This chapter introduced the motion control problem relative to the lower limb exoskeleton robot. Since human gait in stable status is similar to a periodical pattern, two main methods have been introduced to deal with such a control problem, which are the ILC method and the adaptive oscillator method. Both two methods have their disadvantages as well, including response speed, control error, and the ability to deal with nonlinearities. To improve these drawbacks, a novel ILC method is proposed in this thesis.

# Chapter 3

# Preliminaries

The study of the motion control problem of the exoskeleton robots is related to many background preliminaries, including the establishment of the dynamic model for both the exoskeleton robot limbs and the human lower limbs, the main algorithms of iterative learning control, as well as the acquisition of the human gait data. Those background preliminaries are introduced in this chapter. Section 3.1 presents the dynamic model of the human lower limb which has been used in the simulation part of this thesis. Section 3.2 introduces the general principles of the ILC algorithms. Section 3.3 explains the problems related to the acquisition of the human gait data. Section 3.4 summarizes the chapter.

## 3.1 Robotic Model Analysis

This section introduces the fundamentals of the human-exoskeleton modelling. Though have some unique characteristics, the modelling problem for the combination of the human limbs and the exoskeleton robots still has many features in common to general rigid-body robotic models. The analytical methods for general robotic modelling and analysis can still be applied to the exoskeleton robot and the human limbs, which include the kinematic methods and dynamic methods. In this thesis, the dynamics analysis methods have been mainly used, and the dynamic model for the combination of the human lower limb and the exoskeleton robot limb presented in this section will be applied in the following chapters.

### 3.1.1 Kinematic Model

The kinematics of the robots can be defined as a branch of mechanics that deals with the description of the motion of the robots without involving the forces that generate motion. When it comes to the multi-body and joint mechanism of a robot, more specifically an exoskeleton, its kinematics involves analysing the motion of each robot link relative to the reference frame, which includes the analytical description of its movement as a function of time, and the non-linear relationship between the robot end-effector position, orientation, and robot configuration.

Then, In this case, the mobility of a multiple link robot is defined as the number of independent parameters and the position of each link needs to be fully specified [123]. A specific robot configuration is a vector of realisable values for the independent parameters at a certain time slot. However, in the control algorithm study case, due to the kinematic model of the exoskeleton can not take the forces and torques into consideration, though kinematic analysis is still helpful for relative studies, it will be more convenient to adopt the dynamic models for relative motion control algorithm designs.

### 3.1.2 Dynamic Model

The study of the dynamics of an object is a part of the classical mechanics field, which aims to research the objects from the view of motions and the causes of the motions, such as forces. In the case of dynamic researches, there are two main ways to describe the relationship between the motions and the forces, which are the forward dynamics ways and the inverse dynamic ways.

The forward dynamic model of an object is its motions expressed by a function of the forces and torques, which is summarised as follows:

$$\begin{aligned}\ddot{\boldsymbol{p}} &= f(\boldsymbol{F}, \boldsymbol{T}),\\ \dot{\boldsymbol{p}} &= \int \ddot{\boldsymbol{p}} dt,\\ \boldsymbol{p} &= \int \dot{\boldsymbol{p}} dt,\end{aligned} \tag{3.1}$$

where $\boldsymbol{p}$ is the position vector of the object, $\dot{\boldsymbol{p}}$ the velocity vector, and $\ddot{\boldsymbol{p}}$ the acceleration vector. $\boldsymbol{F}$ and $\boldsymbol{T}$ represent the forces and torques that have been applied to the object which lead to the designated motions.

In the mean time, the inverse dynamic model of an object is the forces $\boldsymbol{F}$ and torques $\boldsymbol{T}$ that are needed to generate a designated motion given by the position $\boldsymbol{p}$, velocity $\dot{\boldsymbol{p}}$ and acceleration $\ddot{\boldsymbol{p}}$:

$$[\boldsymbol{F}, \boldsymbol{T}] = g(\boldsymbol{p}, \dot{\boldsymbol{p}}, \ddot{\boldsymbol{p}}). \tag{3.2}$$

Considering a multiple link robot with $N$ joint actuators, its dynamic model can be described as a nonlinear second-order vector differential equation:

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{\tau} - \boldsymbol{\tau_d}, \tag{3.3}$$

where $\boldsymbol{M}(\boldsymbol{\theta}) \in \mathbb{R}^{N\times N}$ is the square inertial matrix which is positive definite. $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}} \in \mathbb{R}^N$ are the joint angle positions, joint angle velocities, and the joint angle accelerations respectively. $\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^N$ represents the centrifugal and Coriolis forces. $\boldsymbol{G}(\boldsymbol{\theta}) \in \mathbb{R}^N$ is the force related to gravity. $\boldsymbol{\tau} \in \mathbb{R}^N$ is the torque serves as control input signal. Finally, $\boldsymbol{\tau_d} \in \mathbb{R}^n$ describes the possible disturbances and noises.

### 3.1.3 Model of the Exoskeleton Robot and Lower Limbs

The overall model for the lower limb of the wearer, after wearing the exoskeleton robot, can be regarded as a combination of the exoskeleton robot model and human lower limb model, which means their dynamics, kinematics, and mechanical structures should be considered together. For the exoskeleton robot aspect, its modelling is straightforward since the structure of the exoskeleton robot has a lot in common with the robotic arms. For the hard-body exoskeleton robot introduced in this thesis, the deformation of its body during the motions are negligible compared to its size, which means its limbs can be regarded as rigid bodies.

The typical structure of an exoskeleton robot is consisted of joints and links, while their numbers depend on the actual design and purpose. Particularly, the structure of the exoskeleton robot studied in this thesis is shown by the side view and front view in Figure 3.1. As shown in the figure, it is a lower limb exoskeleton robot that has seven body links and six joints. The seven body links, which are attached to the corresponding human body segments, include a trunk link, two thigh links, two shank links, and two foot links. $l_1$ is the length of the thigh link while $l_2$ is the length of the shank link. The centre controller and power units are mounted on the trunk link. The six joints contain two hip joints, two knee joints, and two ankle joints respectively. The Degree of Freedom (DoF) designs for the three kinds of joints are different. The hip joints have three DoF to provide a wider Range of Motion (RoM) and more flexibility for the entire legs. The knee joints have only one DoF in order to enhance the structural strength of

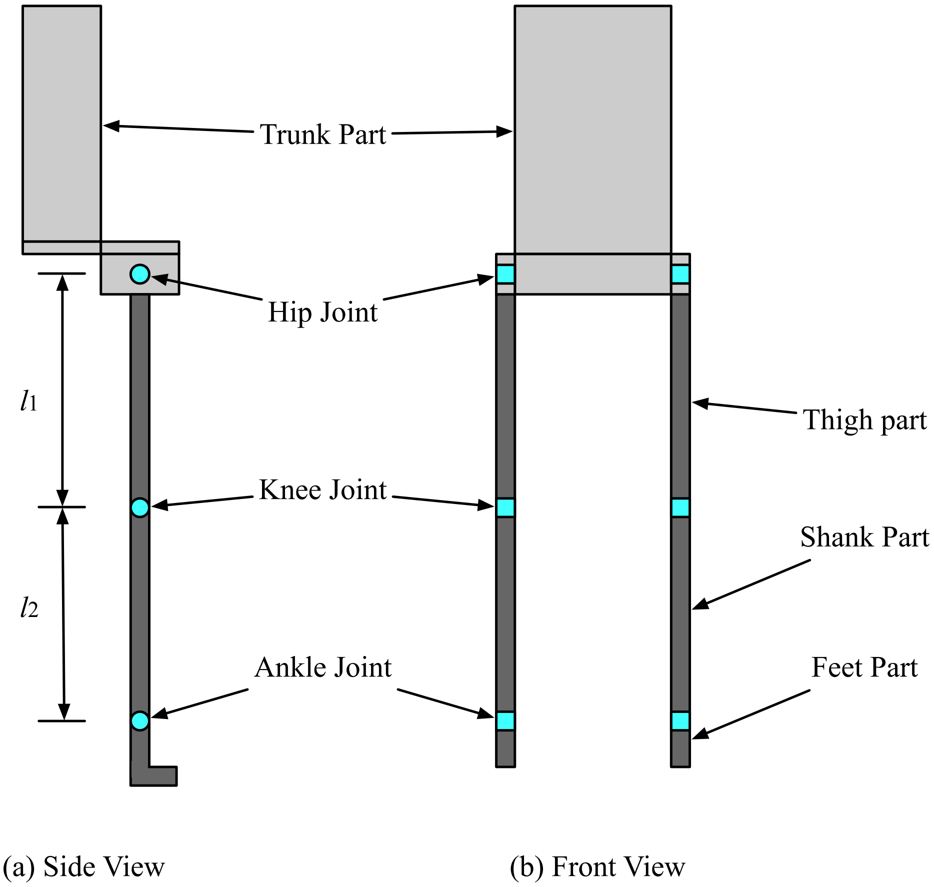

Figure 3.1: Side and Front Views of A Lower Limb Exoskeleton.

the joints and simplify the mechanical design, as the knee joints usually will not twist around the axes of legs during the normal walking process. Meanwhile, the ankle joints also have three DoF to provide more wearing comfort, making up the total DoF of each exoskeleton leg reaching seven.

Among those six joints, what is worth noting is that in rehabilitation cases usually only the knee joints and one DoF of the hip joints of the exoskeleton will be power-assisted by the actuators [53]. For rehabilitation exoskeleton robots, such a design is a balance between performance and system complexity. During the normal walking process of the wearer, the most torques for leg motions are provided by the muscles at the hip, thigh, and shank positions [124]. By offering power assistance to the two DoF of hip and knee joints, it is already enough for the exoskeleton to provide necessary assistant torques for normal walking. The effectiveness of the waking assist will not considerably increase even if the assistant actuators for more DoF have been added. Hence, in order to simplify the motion control problem of the lower limb exoskeletons, it is reasonable to model the exoskeleton legs as a 2-DoF 2-link rigid-body robot.

For the human lower limb aspect, with some reasonable assumptions, the methods used to study the robots can also be applied to the analysis of human kinematics and dynamics. For example, one of the most common assumptions is to assume the human limbs during normal walking gait can also be regarded as rigid-body links. In reality, the human limbs are not right-body, and their deformation in length during normal walking gait is larger than those of the robot limbs. But in exoskeleton cases, such errors are not obvious. In the meantime, after the wearer wears on the exoskeleton robot, the limbs of the wearer and the exoskeleton robot will be in fact tied together. The wearer will stand on the foot links of the exoskeleton robot, and their motions will also be synchronized. Also, at this stage, the main body parts of the wearer that will be focused on are the thigh segments, shank segments, hip joint, and knee joints, as the trajectory of human lower limbs during normal walking is mainly determined by them. Hence, it is reasonable to describe the dynamics of the human lower limbs via a 2-link rigid-body model, where the feet segments are regarded as extensions of the shank segments in length. The segments of the model are connected by joints that mimic the human joints in terms of DoF and RoM. The dynamics of the ankle joints are not considered in this thesis to simplifying the problem, as usually they are also not power-assisted by the actuators in real designs.

For such a model, the relative anthropometric data needed are shown in Table 3.1 [125]. In this table, the weight and length for the corresponding limb segments are given in relative percentages, where M represents the mass of the entire human body, P represents the distance is calculated from the proximal end of the segment, and H represents the height of the entire human body.

Table 3.1: Anthropometric Data of Human Lower Limb Segments

| Segment | Segment Weight / Total Body Weight | Centre of Mass Position / Segment Length | Segment Length/ Total Body Height |
|---|---|---|---|
| Foot | 0.0145 M | 0.50 P | 0.039 H |
| Shank | 0.0465 M | 0.433 P | 0.246 H |
| Thigh | 0.100 M | 0.433 P | 0.245 H |
| Foot and Shank | 0.061 M | 0.606 P | 0.285 H |
| Total Leg | 0.161 M | 0.447 P | 0.530 H |
| Pelvis | 0.142 M | 0.105 P | 0.045 H |

Hence, in this thesis, the final model for the combination of the human lower limb and the exoskeleton robot is a 2-DoF 2-link rigid-body model, which is shown in Figure 3.2. Though the total DoF of the original combination is more than two, only the two power-assisted DoF are considered as they are the major concerns. As demonstrated in

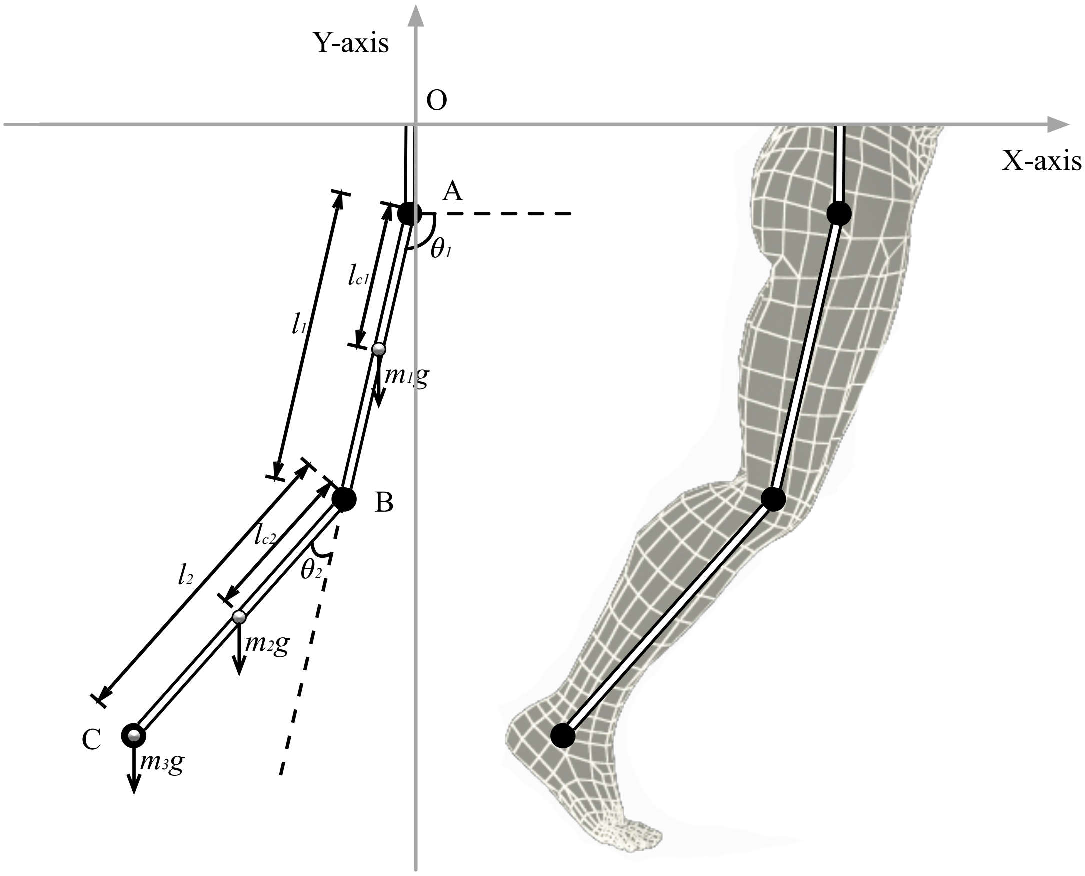

Figure 3.2: Overall 2-Link Model of Lower Limbs.

the figure, the thigh part and shank part of the combination is modelled as a rigid-body link. According to Table 3.1, the Centre of Mass (CoM) for the foot, shank, and thigh segments are all very close to the middle of its total length. Hence, it is reasonable to regard the links of the rigid-body model as homogeneous links, whose CoM are located in the middle of each link. The foot segment is regarded as a mass point attached to the distal end of the second link, while the distance from its CoM to the ankle joint has been added to the length of the link.

As demonstrated in Figure 3.2, $l_1$ and $l_2$ are the lengths of the first and second links, which represent the lengths of the thigh segment and shank segment of the exoskeleton robot respectively. $l_c1$ and $l_c2$ are the distances between the CoM and the proximal end of the segments, which equal to half of the segment length $l_1$ and $l_2$. $m_1$, $m_2$ and $m_3$ are the weights of the thigh, shank, and foot segments respectively. $\theta_1$ is the hip joint angle while $\theta_2$ is the knee joint angle. The two joint actuators of each leg is mounted on the hip joint and knee joint, corresponding to angle $\theta_1$ and $\theta_2$.

In terms of the specific 2-DoF 2-link rigid-body model, its dynamic model can still be described by the nonlinear equation (3.3), while the terms $\boldsymbol{\theta}$, $\boldsymbol{\tau}$, $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{G}$ of the equation are expressed as follows:

The angle vector $\boldsymbol{\theta}$ defined by:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \tag{3.4}$$

represents the joint angles for hip joint and knee joint.

The torque vector $\boldsymbol{\tau}$ defined by:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}, \tag{3.5}$$

represents the joint torques generated by the hip and knee joint actuators of the exoskeleton robot. Such torques serves as the control input signals to the 2-link model system.

Afterwards,

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \tag{3.6}$$

is a positive definite square matrix which represents the inertial of the system, where $m_{11}, m_{12}, m_{21}, m_{22}$ are the elements of the square inertial matrix. These elements are expressed by:

$$\begin{aligned} m_{11} &= m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos\theta_2) + m_3(l_1^2 + l_2^2 + 2l_1 l_2 \cos\theta_2), \\ m_{12} &= m_2(l_{c2}^2 + l_1 l_{c2} \cos\theta_2) + m_3(l_2^2 + l_1 l_2 \cos\theta_2), \\ m_{21} &= m_{12}, \\ m_{22} &= m_2 l_{c2}^2 + m_3 l_2^2. \end{aligned} \tag{3.7}$$

Then,

$$\boldsymbol{C} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, \tag{3.8}$$

represents the effects of the centrifugal and Coriolis forces to the system, whose elements $c_{11}, c_{12}, c_{21}$, and $c_{22}$ can be obtained by:

$$\begin{aligned} c_{11} &= h\dot{\theta}_2, \\ c_{12} &= h\dot{\theta}_1 + h\dot{\theta}_2, \\ c_{21} &= -h\dot{\theta}_1, \\ c_{22} &= 0, \end{aligned} \tag{3.9}$$

among which $h$ is given by:

$$h = -m_2 l_1 l_{c2} \sin\theta_2 - m_3 l_1 l_2 \sin\theta_2. \tag{3.10}$$

And finally,

$$\boldsymbol{G} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}, \tag{3.11}$$

is the gravity matrix, where $G_1$ and $G_2$ are the gravity elements formulated by:

$$\begin{aligned} G_1 &= (m_1 l_{c1} + m_2 l_2) g \cos\theta_1 + (m_2 l_{c2} + m_3 l_2) g \cos\theta_1 + \theta_2, \\ G_2 &= (m_2 l_{c2} + m_3 l_2) g \cos\theta_1 + \theta_2. \end{aligned} \tag{3.12}$$

In this research, it is assumed that there is no input noise in the framework, which leads to:

$$\boldsymbol{\tau_d} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3.13}$$

As a summary, this section introduced the kinematic and dynamic models of a robot system. Then, by a series of assumptions, the dynamics model of the system has been given for further ILC algorithm designs.

## 3.2 Iterative Learning Control

After explaining how to establish the dynamic model for the rehabilitation exoskeleton robot, this section introduces the basics of ILC methods, the typical ILC algorithms for linear and nonlinear systems, as well as the relative design criteria for ILC systems. They are the main control methods that have been studied in this thesis.

### 3.2.1 Background

As introduced in Section 2.2.1, the study of the ILC algorithm begins in 1984 [82] and has been widely applied in various fields since the 2000s [126]. In recent years, a number of researches about its applications for more complicated systems and scenarios have also been proposed, for example for input constraints in 2017 [127] and for cooperation with neural networks in 2020 [128]. In principle, ILC algorithms mainly focus on systems that have repetitive or periodic procedures, for example, the repetitive control of servomotors. The entire control procedure of the system can be divided into a series of time intervals, where the length of each time interval is equal to the time length of each repetitive period. The length of the time intervals is fixed and finite. Then, the system is assumed to be invariant between those time intervals, which means that the states of the system will share the same initial value at the start point of each time interval, and the dynamics of the system will also be the same in each interval if not been interfered by external signals. The application of the ILC algorithm does not require full knowledge of the system. It also does not focus only on the current interval itself. In each time interval, the ILC algorithm will make use of the knowledge and experiences the system obtained from previous intervals. The output errors of the system in previous intervals will help in constructing the control input in the next time interval. Finally, as time goes by, the number of these time intervals will increase and the output error of the system will generally convergent to zero [129]. Compared to other learning algorithms based on neural networks [130–132], the ILC algorithm is much simpler while the entire learning process is completed online.

These notable features have led to the establishment of ILC as an interesting and ongoing area for control system researches and applications. In recent years, more theoretical works have been done by the researchers, such as [86, 133–135], but the concept of learning from the previous knowledge to improve future performance has been retained. Also, during the development of the ILC algorithm, the idea of ILC has been expanded in both width and depth, including intersect with other control topics such as adaptive control [136], robust control [137], and optimal control [138]. The applications of ILC methods have also been broadened beyond the industrial robotics and process control field [139].

For the novel ILC algorithm proposed in this thesis, which combines the ILC method with a feedback linearization method, the relative background preliminaries are introduced as follows.

### 3.2.2 Iterative Learning Control for Linear Systems

In this section, the ILC algorithms for linear systems is introduced. Firstly, the principle of ILC control law design is given. Then, three common used types of ILC algorithm are presented, which are P-Type, PD-Type and NOILC algorithms. Due to the vigorous development of ILC algorithms in recent decades, there still exist many other types of ILC algorithms, including the model inversion ILC and gradient descent ILC algorithm, but they will not be detailed introduced in this section.

For a Single-Input Single-Output (SISO) or Multi-Input Multi-Output (MIMO) discrete linear time-invariant system with $m$ states, $p$ inputs and $q$ outputs, its state-space form model is given by:

$$\begin{aligned} x(i+1) &= Ax(i) + Bu(i), \quad i = 0, 1, 2, \cdots, \quad x(0) = x_0, \\ y(i) &= Cx(i) + Du(i), \end{aligned} \tag{3.14}$$

where $x$ is the state, $y$ is the output, $u$ is the control input, $i$ is the discretized time variable, $A$ is an $m \times m$ system matrix, $B$ is an $m \times p$ input matrix, $C$ is a $q \times m$ output matrix, and $D$ is a $q \times p$ transfer matrix. Without losing generality, it is assumed $D = 0$ if not specially mentioned.

In ILC cases, the entire dynamics of the system is divided into a series of fixed and finite time intervals. Each interval is regarded as an iteration, whose length is decided by the period length of the system. After each iteration ends, the system will be reset to the initial status and start a new iteration. In $k$th iteration of ILC, the system model can be writen as:

$$\begin{aligned} x_k(i+1) &= Ax_k(i) + Bu_k(i), \quad i = 0, 1, 2, \cdots N-1, \quad x_k(0) = x_0, \\ y_k(i) &= Cx_k(i) + Du_k(i), \end{aligned} \tag{3.15}$$

where $N$ is the number of sample points in each iteration.

For a certain reference signal $r(i)$, the error between the system output and the reference in $k$th iteration is given by:

$$e_k(i) = r(i) - y_k(i). \tag{3.16}$$

In order to rewrite the system in a more compact form, the following vectors are introduced:

$$\begin{aligned} y_k &= [y_k(0),\ y_k(1),\ \cdots,\ y_k(N-1)]^T, \\ u_k &= [u_k(0),\ u_k(1),\ \cdots,\ u_k(N-1)]^T. \end{aligned} \tag{3.17}$$

Then, the system model can be written as:

$$y_k = Gu_k, \tag{3.18}$$

where $G$ is a $Nm \times Np$ lower triangular Toeplitz matrix [140]:

$$G = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-2}B & CA^{N-3}B & CA^{N-4}B & \cdots & D \end{bmatrix}. \tag{3.19}$$

Then, the control objective of the ILC algorithm is, as the iteration time $k$ increases, the output error should converge to zero. By introducing the compact form of reference signal and system error as well:

$$\begin{aligned} r &= [r(0),\ r(1),\ \cdots,\ r(N-1)]^T, \\ e_k &= [e_k(0),\ e_k(1),\ \cdots,\ e_k(N-1)]^T, \end{aligned} \tag{3.20}$$

the control objective of the ILC algorithm can be written as:

$$\lim_{k\to\infty} \|e_k\| = 0, \tag{3.21}$$

$$\lim_{k\to\infty} \|u_k - u_\infty\| = 0, \tag{3.22}$$

where $u_\infty$ is the converged input sequence. The converged input sequence in ILC cases means it is the value of the input signal when it finished its learning progress and converged to a steady-state status. Since $u_\infty$ is not a known parameter that can be defined before the start of the ILC algorithm, in real applications, whether objective (3.22) has been fulfilled is judged by monitoring the value of $u_k$. If the value of $u_k$ has not been changing for a long enough period while the control error of the ILC algorithm converges, objective (3.22) can be regarded as fulfilled.

The value of the conveged input $u_\infty$ is decided by the final value of the learning process of the controller. The learning process is given by:

$$u_{k+1} = u_k + \Delta(u_k, e_k), \tag{3.23}$$

where $\Delta(u_k, e_k)$ is the correction term. It is a function of the previous control signal and error signal. It represents how much the controller has been modified between each iteration.

Finally, if the output disturbance of the system is taken into account, system output (3.18) can then be rewritten as:

$$y_k = Gu_k + d, \tag{3.24}$$

where $d$ is the output disturbance defined by:

$$d = [d(0),\ d(1),\ \cdots,\ d(N-1)]^T. \tag{3.25}$$

To obtain a better control performance, one of the rules in ILC control law design is to make sure the dynamics of the system error $e_k$ is asymptotically stable. If this requirement is not fulfilled, an inner loop controller of the system should be designed in advance in order to stabilize the system error [141]. Several common used ILC control laws are given in the following sections.

### D-Type ILC

By taking a derivative on system error, a D-Type ILC control law is given by:

$$u_{k+1}(i) = u_k(i) + K_d(e_k(i+1) - e_k(i)), \tag{3.26}$$

or in compact form:

$$u_{k+1} = u_k + K_d\dot{e}_k, \tag{3.27}$$

where $K_d$ is the learning coefficient. The system error will converge to zero with D-Type ILC control law if and only if $\|I-CBK_d\| < 1$, or in other words, discrete system matrix $I - CBK_d$ is stable [133]. It is worth noting that the criteria for the performance of ILC systems will be slightly different from that of standard control systems, as the system dynamics over the iteration numbers have to be taken into account as well.

A interesting feature of this D-Type ILC control law is that the convergence of system error does not depend on the system matrix $A$. This is due to the time length of each iteration is finite and the system is linear. Even if the original system (3.26) is unstable, the system error can still convergence as iteration number increases. However, in this case, the performance of the ILC controller and the error convergence speed will be decreased due to the instability.

### PD-Type ILC

Based on the D-type ILC control law (3.26), there are many variations including the P-Type, PI-Type, PD-Type and PID-Type ILC control law, among which the most commonly used one is the PD-Type ILC control law. The PD-type ILC control law is given by:

$$u_{k+1}(i) = u_k(i) + K_p e_k(i+1) + K_d(e_k(i+1) - e_k(i)), \tag{3.28}$$

or in compact form:

$$u_{k+1} = u_k + K_p e_k + K_d \dot{e}_k, \tag{3.29}$$

where $K_p$ is the proportional learning coefficient and $K_d$ is the derivative learning coefficients. In some cases, the proportional item $e_k(i+1)$ in (3.28) may also be replaced by $e_k(i)$ or $e_{k+1}(i)$ [90, 142].

For systems with the PD-type ILC control law, the system error convergence if and only if $\|I - CB(K_p + K_d)\| < 1$, or in other words, discrete system matrix $I - CB(K_p + K_d)$ is stable [85]. As the structure of the PD-type ILC controller is inspired by the PD feedback controller, the effects of $K_p$ and $K_d$ on the system performance also have similarity to some extent. Larger learning coefficients will lead to a speed up in convergence rate, but will also risk in causing strong oscillation on system error. Smaller learning coefficients will suffer from a reduction of error convergence rate [143]. Usually, by carefully tuning the learning coefficients $K_p$ and $K_d$, PD-type ILC control law can have a better tracking performance and error convergence rate, compared to other PID-form ILC control laws.

### Norm Optimal ILC

The concept of norm optimal iterative learning control (NOILC) is based on the gradient theory. It combines the ILC algorithm and optimal control in order to achieve a better control performance.

From the discussions in previous sections, it is obvious that for a system applying the ILC algorithm, its entire dynamics can be divided into two layers. The first layer is the system dynamics by time in each single iteration, while the second layer is the system dynamics between different iterations. The overall error convergence performance of the system depends on the dynamics of both two layers. Then, the basic idea of NOILC is to decouple the control design of those two layers so that they can be designed and optimized separately.

NOILC introduces a independent criteria to judge the performance of the system based on the input signal and output error of each iteration. In this way, the design of the system dynamics on the iteration layer is no longer directly related to the variable of time. The design can now be transformed into an optimal problem and can be solved in optimal ways. In each iteration, the NOILC algorithm is a combination of both feedback control and feedforward control. It takes the error data from the current iteration as a feedback loop, while the data from the previous iterations serve as a feedforward loop [135, 144].

The main idea of the NOILC algorithm is given by:

$$u_{k+1} = \arg\min_{u_{k+1}} \|r - y_{k+1}\|_Q^2 + \|u_{k+1} - u_k\|_R^2, \tag{3.30}$$

where $Q \in \mathbb{R}_{++}^{N\times N}$ and $R \in \mathbb{R}_{++}^{N\times N}$ are optimal weight matrices that must be symmetric and positive definitive. The front quadratic item $\|r - y_{k+1}\|_Q^2$ represents the system error that decides the tracking performance of the system. The latter quadratic item $\|u_{k+1} - u_k\|_R^2$ is a penalty item that represents the changes in the input signal. By taking such changes into the optimal calculation, the system stability can be enhanced while the convergence performance can also be improved.

By reformulating the NOILC algorithm (3.30) to an optimal control form, the NOILC problem can be converted to an optimization problem, whose cost function is given by:

$$J(u_{k+1}) = \|r - y_{k+1}\|_Q^2 + \|u_{k+1} - u_k\|_R^2, \tag{3.31}$$

or equivalently:

$$\begin{aligned} J(u_{k+1}) &= \sum_{i=0}^{N-1}(r - y_{k+1}(i))^T Q(r - y_{k+1}(i)) \\ &+ \sum_{i=0}^{N-1}(u_{k+1}(i) - u_k(i))^T R(u_{k+1}(i) - u_k(i)) \end{aligned} . \tag{3.32}$$

To achieve the optimal status, the cost function (3.31) must fulfill the stationarity condition that:

$$\frac{\partial J(u_{k+1})}{\partial u_{k+1}} = 0. \tag{3.33}$$

By substituting the cost function (3.31) and the system model (3.18) into condition (3.33), there is:

$$-G^T Q(r - Gu_{k+1}) + R(u_{k+1} - u_k) = 0. \tag{3.34}$$

By reformulating the equation with the system error (3.16), the NOILC control law can be given by:

$$u_{k+1} = u_k + R^{-1} G^T Q e_{k+1}. \tag{3.35}$$

The system error convergent condition of this control law is either the kernel of matrix $G^T$ is equal to zero or $r \in \mathrm{range\ G}$ [135].

However, one remaining problem of the NOILC control law (3.35) is that such a control law is not causal. In normal cases, the system error of the $k+1$th iteration $e_{k+1}$ is impossible to be obtained if the corresponding system input $u_{k+1}$ has not been decided yet. But in NOILC cases, it is possible since the system error can be calculated in another way. By introducing a familiar costate system on a reverse time sequence from $i = N-1$ to $i = 0$, there is:

$$\begin{aligned} &\phi_{k+1}(i) = A^T \phi_{k+1}(i+1) + C^T Q(t+1) e_{k+1}(i+1), \\ &u_{k+1}(i) = u_k(i) + R^{-1}(i) B^T \phi_{k+1}(i), \end{aligned} \tag{3.36}$$

where $N-1 \geq i \geq 0$, and the terminal condition of the costate system is given by $\phi_{k+1}(N) = 0$.

Assuming the states of the anticausal costate system (3.36) is fully known, a causal implementation can be found for it in [145]. Then, the costate system (3.36) can be transformed to:

$$\phi_{k+1}(i) = -K(i)(I + BR^{-1}B^TK(i))^{-1}A(x_{k+1}(i) - x_k(i)) + \xi_{k+1}(i), \tag{3.37}$$

where $\xi_{k+1}(i)$ is a feedforward item and $K(i)$ is a gain matrix.

The value of $K(i)$ is given by the solution of the corresponding discrete Riccati equation of the costate system (3.36) [146], which is given by:

$$\begin{aligned} K(i) &= A^TK(i+1)A + C^TQC \\ &\quad - A^TK(i+1)B(B^TK(i+1)B + R)^{-1}B^TK(i+1)A, \end{aligned} \tag{3.38}$$

where $N-1 \geq i \geq 0$, and the terminal condition of $K(i)$ is given by $K(N) = 0$. It is worth noting that the value of $K(i)$ is only dependent on the original system (3.14) as well as the optimal weight matrices $Q$ and $R$. It is not related to the control input, system output, status, and error signals in each iteration, which means its value can be calculated in advance of each control progress.

For the remaining unknown value in the costate system (3.36), the feedforward item $\xi_{k+1}(i)$ in the equation is obtained by:

$$\xi_{k+1}(i) = (I + K(i)BR^{-1}B^T)^{-1}(A^T\xi_{k+1}(i+1) + C^TQe_k(i+1)), \tag{3.39}$$

where the terminal condition is also given by $\xi_{k+1}(N) = 0$. The value of $\xi_{k+1}(i)$ is related to the value of the gain matrix $K(i)$ and the system error $e_k(i+1)$. $K(i)$ is a constant during each control progress and $e_k(i+1)$ can be obtained by the tracking error of previous iterations.

Since in solving the optimization problem for NOILC algorithm (3.30) a model inversion method has been applied in the procedure to calculate relative coefficients, some also regard NOILC not a completely different method, but a general variation of implementing model inversion ILC [147].

Compared to other commonly used ILC algorithms, the major advantages of the NOILC algorithm lies in error convergence performance, including convergence speed, tracking error, and the monotonicity of convergence. Such properties lead to a wide range of NOILC applications for robotics, manipulators, Unmanned Aerial Vehicle (UAV)s, and exoskeletons [148, 149].

In the meantime, the NOILC algorithm also has its disadvantages. For example, when dealing with a system with nonlinearities, the dynamics of the system can be much complex than the linear system in (3.14). The optimal problem for cost function (3.31) will include nonlinear least-squares items, which is non-convex and a global optimal cannot be guaranteed [150, 151]. To deal with the nonlinear ILC problems, either some linearization method should be applied to linearize the system and improve the performance of NOILC, or different ILC methods shall be used.

### 3.2.3 Iterative Learning Control for Nonlinear Systems

The majority of ILC studies in the past few decades are focused on linear objects. However, due to the wide existence of nonlinearities in both industrial practice and daily environment, ILC algorithms for nonlinear systems has also been widely studied in the literature [152–154].

A typical example of a nonlinear system is the manipulator system, which can be commonly seen in industrial practice. Its dynamics can be expressed by the following model based on torques:

$$M_r(x(t))\ddot{x}(t) - C_r(x(t),\dot{x}(t))\dot{x}(t) - g_r(x(t)) - d_r(x(t),\dot{x}(t)) = \tau(u(t)), \tag{3.40}$$

where $\ddot{x}(t), \dot{x}(t), x(t)$ are the accelerations, velocities, and positions of the manipulator limbs, respectively. $M_r$ is the inertial matrix, which have to be positive definite. $C_r$ represents the effects of Coriolis force. $g_r$ is the force vector of gravity. $d_r$ represents the vector of friction torque and other disturbances. Finally, $\tau$ is the output torque of actuators. which is generated by the control input $u(t)$.

One way of categorizing nonlinear systems is to judge whether the control of the system is affine or not. An affine system can be formed as:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + B(x(t))u(t), \\ y(t) &= h(x(t)), \end{aligned} \tag{3.41}$$

while a non-affine system can be formed as:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + B(x(t)u(t)), \\ y(t) &= h(x(t)), \end{aligned} \tag{3.42}$$

where $x, u, y$ are the state vector, input, and output of the system while $f, B, h$ are nonlinear functions.

Another way of categorizing the nonlinear systems is via their continuity, which leads to Global Lipschitz Continuous (GLC) systems, Local Lipschitz Continuous (LLC) systems, and other systems [153]. Whether a function is Lipschitz continuous or not depends on whether there exists a non-negative real number such that, for any part of the designated function, the absolute value of its derivative should not be greater than this number. The smallest value of such number is named as Lipschitz constant. In recent years, GLC systems and LLC systems have been the main attractions for nonlinear ILC researches.

Besides, the convergence proof of the system errors dependent on the iteration numbers has became a very important direction of nonlinear ILC studies as well. Relative system dynamics by iterations have also been studied.

For affine systems, an important condition for applying ILC algorithms is the Lipschitz continuous condition. for the affine system in the form of (3.41), its Lipschitz continuous condition is given by:

$$\begin{aligned} |f(x_1) - f(x_2)| &\leq f_0|x_1 - x_2|, \\ |B(x_1) - B(x_2)| &\leq b_0|x_1 - x_2|, \\ |h(x_1) - h(x_2)| &\leq h_0|x_1 - x_2|, \end{aligned} \tag{3.43}$$

where $f_0, b_0, h_0$ are Lipschitz constants. If Lipschitz continuous condition (3.43) holds for any $x_1$ and $x_2$ in the domain of system (3.41), system (3.41) will become a GLC system. Otherwise, if for any $x$ in the domain of system (3.41), the Lipschitz continuous condition only holds for a neighborhood $U$ of $x$, then system (3.41) will become a LLC system.

There are many possible ways to deal with the ILC problems for nonlinear systems. One of the ways is to apply the linear ILC algorithms to nonlinear systems, which includes using linear ILC methods directly to nonlinear systems that can fulfill the GLC conditions, as well as treating the nonlinearities in nonlinear systems as a particular kind of noise that has been added to the linearised system models.

In these cases, the output tracking performance will become the main target of ILC control. The relative degree of the system is usually assumed to be zero, and the state information of the system is usually estimated. To make sure the system error can converge in a finite time interval, an identical resetting condition is also required, which means the initial condition of the system in each iteration should be the same and equal to the initial condition of the target trajectory [134]. However, sometimes this condition can be difficult to fulfill in practical applications.

The other way to deal with the nonlinear ILC problems is to design new nonlinear ILC algorithms for nonlinear systems which can fulfill the GLC conditions or LLC conditions. Nonlinear ILC design for LLC systems has become one of the recent topics in nonlinear ILC studies. In these cases, the state information of the system is assumed to be available, while the relative degree of the system is assumed to be equal to or more than one [153]. To solve these problems, several nonlinear ILC algorithms have been proposed, including the Lyapunov-function based Composite Energy Function (CEF) methods, adaptive ILC methods for unknown or time-varying parameters, robust ILC methods for system uncertainties, internal model based ILC methods, and function approximation ILC methods.

Different from theoretical analysis, in the real application environment, most of the controllers are implemented based on discrete-time systems. Though sometimes the discrete-time model for a nonlinear system will be difficult to build, it is still possible to design a discrete time ILC control algorithm based on the continuous-time model of the system. By implementing such ILC control algorithms, the output error of the system will be easier to measure. In this way, the analysis of the error convergence over iteration numbers and the study of implementing the ILC algorithms to nonlinear systems can still be completed.

### 3.2.4 Design Criteria

While designing an ILC algorithm for the objective system, the choice of design criteria is also of great importance. Different design criteria preferences will lead to different strategies and designing focuses on ILC controller designs [85]. In the meantime, compared to the design criteria for normal control algorithms, there are certain criteria that are unique in ILC cases. In this subsection, four main criteria of ILC controller design are introduced, which are stability, performance, transient error performance, and robustness.

### Stability

Obviously, stability analysis is an indispensable and important part of control theory. In most cases, the controller has to consider its stability before its design. In iterative learning control, there are two dimensions of system stability that have to be taken into consideration. The first dimension is the system stability in the scale of time inside each iteration. The second dimension is the system stability in the scale of iteration numbers, which decides the stability of the system during the iteration progress. The overall stability of the ILC system is decided by both two dimensions of stabilities.

For the first dimension of system stability, at first glance, its criteria are similar to the stability of ordinary control problems. But the real difference between them is, due to the time length of each iteration is finite, the system response will not go to a real infinite value in such a finite time scale. In this sense, the system will not be real unstable in these finite iterations.

However, such a condition is not enough for the stability design of the ILC algorithm inside each iteration. In practical applications, there will always be physical limitations for the control objects. For example, motors have their maximum rotating speeds and output torques. In exoskeleton cases, the rotating angle of each joint actuator on the exoskeleton must also be limited in order to guarantee the safety of the wearer. Though the system response is finite in each finite iteration, it may still break the physical limitations of the control objects.

Hence, the system stability inside each iteration should still be noticed and taken into consideration in ILC controller designs. It is better to make sure the system response will not break those limitations in any iteration.

The second dimension of system stability, which is the system stability over iteration numbers, is defined in the input space. Compared to the previous dimension, such stability is unique in ILC designs. To achieve this dimension of stability, the input sequence in each iteration should asymptotically converge to a converged input sequence [85, 155], which can be written as:

$$\exists u_\infty : \lim_{k\to\infty} \|u_\infty - u_k\| = 0, \tag{3.44}$$

where $u_\infty$ is the converged input sequence. As explained previously in subsection 3.2.2, it represents the value of the input signal when it finished its learning progress and converged to a steady-state status.

Also, it is worth noting that the system stability over iteration numbers does not guarantee a convergence for the output error of the system. It only represents when the iteration number increases, the output error will always be bounded. The convergence of the output error is categorized in the performance part.

### Performance

Except for the stability, system performance is also a vital criterion for ILC controller designs. The exact benchmark of system performance usually varies with its physical properties and designing target. But in general cases, the performance of a control system will refer to its tracking performance, which is, for a certain given reference signal, how will the system act and follow such a reference.

In ILC problems, the tracking performance of the system is represented by the output error, which is the difference between the system output and the desired reference signal. Just like the stability criterion, the output error of the system also has to be considered in two dimensions, the first one is the error inside each iteration, while the second is the error over iteration numbers. There are many ways to measure and compare the tracking performance between different controllers and algorithms, one of which is to compare the Euclidean norms of the initial errors and the errors when iteration numbers go to infinite, which is $\|e_0(i)\|_2$ and $\|e_\infty(i)\|_2$. The convergence speed of error will also be considered. In practical applications, the true value of the error when iteration numbers go to infinite will be hard to obtain, so usually the value of the final iteration error will be regarded as the value of $\|e_\infty(i)\|_2$.

It is also worth noticing that the stability of the system is not strictly related to the tracking performance, which means the system does not have to be stable to obtain a good tracking performance, or in other words, the output error does not have to be convergence for the entire period of ILC process. In some cases of ILC controller design, there exist a phenomenon that the output error of the system will converge to a small value in the first several iterations of the ILC process, then goes to unstable in the following part as the iteration number increases. For normal control problems, such instability is usually unacceptable. But in ILC control cases, it is possible for the system to run normally for the first several iterations, then cut off the ILC controller and stop the iterative learning process before the output error goes unstable. The controller can be kept in the status when the output error is the smallest. In this way, good tracking performance of the system can still be achieved even if the system is unstable and the error output is not globally converging.

In the meantime, there are also cases that even when the system is stable and the output error finally converges, the ILC algorithm is still unacceptable. Such cases are due to the transient error problem unique in ILC control, which will be introduced in the next subsection.

### Transient Error Performance

The transient error performance is a criterion specific for ILC controllers. Such a criterion is set in response to the transient error problem that happens during ILC controller designs. For the ILC controllers, there exists a phenomenon that sometimes even for stable ILC systems with convergent output errors, their error convergence processes may still be non-monotonic. The output error will increase rapidly in the first few iterations of the ILC process, then gradually decrease, and finally, converge to an expected bound. Though the final norm value of the output error $||e_\infty(i)||_2$ may still meet the performance criteria, the peak value of the error can be harmful to the system, causing damage to the controllers or actuators. Such problems should be avoided during the ILC controller designs.

A typical example of the transient error problem over iteration numbers in ILC designs is shown in Figure 3.3. It is obtained while the ILC algorithm proposed in this thesis is being simulated. In the figure, the $x$-axis represents the iteration numbers while the $y$-axis represents the average absolute output error value of each iteration during the ILC process.

As demonstrated in the figure, in the first 12 iterations, the output error has encountered a tremendous rise from near zero to a peak value of more than 2000. Then between iteration 12 and iteration 50, the output error has faced a reduction with a bit of oscillation. After iteration 50, the value of the output error has been gradually settled to a relatively small scale again. Though in this case the system is asymptotically stable and the output error converges, such a scene is definitely not desired in ILC controller designs. In order to evaluate and analyze systems with such problems, the transient error performance

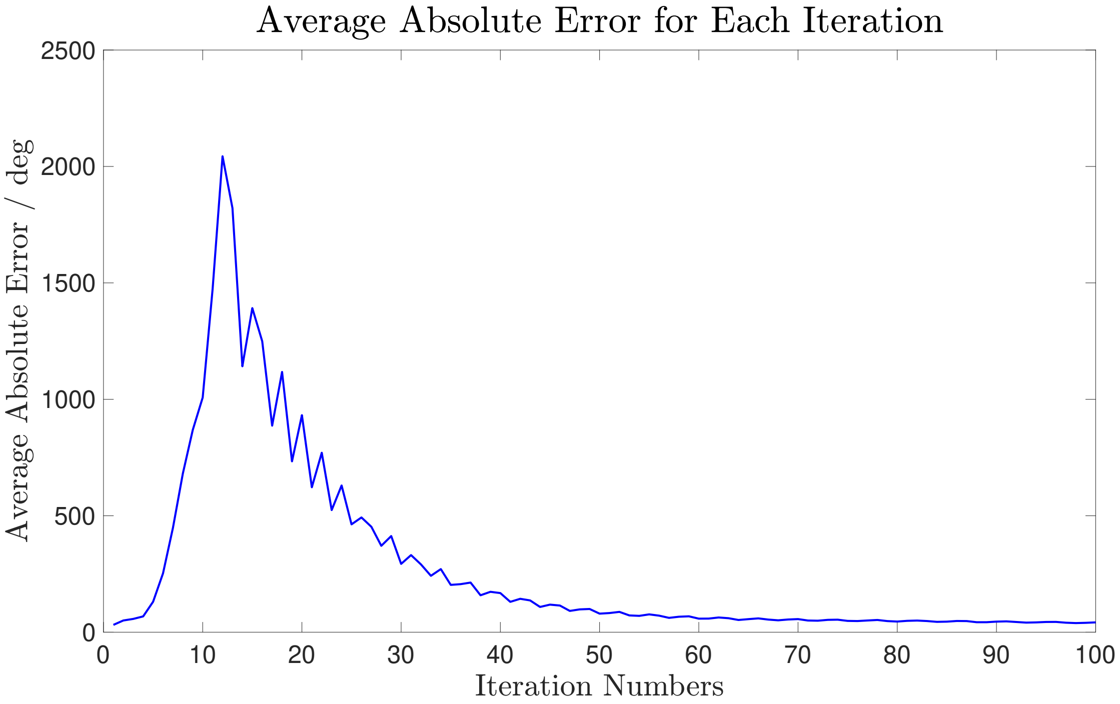

Figure 3.3: An Example of the Transient Error Problem.

criterion is introduced.

To avoid the transient error problem of ILC systems, one of the important features is that the convergence of the system output error should be monotonic. For both linear system (3.14) and nonlinear system (3.41), their monotonic error convergence condition can be written as [156]:

$$\|e_\infty - e_{k+1}\| \leq \lambda \|e_\infty - e_k\|, \tag{3.45}$$

where $e_\infty$ is the converged error when the iteration number goes to infinity, and $\lambda$ is the error convergence rate whose range is given by $0 \leq \lambda < 1$.

For linear ILC systems, the transient error problem is usually related to the learning gain of the proposed ILC algorithm. A larger learning gain will have a higher possibility of leading to the transient error problem. For nonlinear ILC systems, the appearance of the transient error problem is not only related to the learning gain but also related to the nonlinear dynamics of the system. Hence, to achieve the monotonic convergence property of the output error, the learning gain of the ILC algorithms must be carefully selected. In particular, for the nonlinear ILC problems, the effects of the nonlinearities on system performance must be considered in advance.

### Robustness

Except for the three main criteria introduced in previous subsections, the last criterion for the ILC controller design is robustness. Similar to those of all other controllers, robustness in the field of ILC also refers to the ability of the controller to deal with the unexpected deviations of the system. Those unexpected deviations include uncertain parameters, time-varying parameters, measurement errors, time delays, unmodelled dynamics, system mismatches, internal and external disturbances, as well as other possible deviations. Though been listed here as the fourth criterion, the robustness of a system is not an independent vector. It is usually represented via other criteria, for example, the stability, error convergence, or tracking performance of the system after adapting the previous unexpected deviations.

Nevertheless, for ILC algorithms, their robustness is still of great importance. Since one of the advantages of ILC methods is the ability to deal with a certain level of uncertainties. ILC methods usually do not require full knowledge of the system. Though some information might be unknown to the controller, the controller can still have a good control performance after a number of iterations. Also, ILC methods are suitable for time-varying systems as well, as the controller can gradually learn the variation of the system and adapt the control protocol correspondingly. Relative studies can be found in [98, 157, 158].

Specifically for ILC systems, the external disturbance signal also has two versions, the first one is that the external disturbances will only appear once in one of the iterations, while the second one refers to external disturbances which will appear continuously in every iteration. Compared to other control methods, ILC methods have advantages in dealing with the second type of disturbances. Due to such external disturbances will appear repetitively in multiple iterations, the control sequence of later iterations can have the experience and knowledge from previous iterations. A feedforward control will then be conducted to eliminate the effect of such disturbances.

In the meantime, ILC algorithms have disadvantages in dealing with the first type of external disturbances. The reason is that in such cases, the disturbance signal will appear only once in one of the iterations, but the system response in later iterations will still be affected by this disturbance. It will cost several successive iterations for the effect to be eased.

As a summary of the previous subsections, it is the combination of the four criteria mentioned above, stability, performance, transient error performance, and robustness, that have constructed a reasonable performance evaluation system for ILC algorithms.

## 3.3 Gait Data

Besides the ILC algorithms and the robotic model of the exoskeleton system, another essential aspect in exoskeleton research is the study of human gait. From the individual point of view, each single step for each individual person is unique. Due to the random disturbance in environment and the chaotic property exists in nature, there will never be any specific step that can be precisely repeated. However, from the statistic point of view, by ignoring some minor errors, the gaits of the human beings are highly periodic, repetitive, and share a number of characteristics in common. This section generally introduces these characteristics of the human gait, as well as how the gait data used in this thesis is obtained.

### 3.3.1 Characteristics of Human Gait

The gait of human beings is a highly ordered process regulated by the central nervous system, while its control is a combination of an autonomous system and the will of human beings. Its rhythm is decided by the Central Pattern Generator (CPG), which is a kind of biological neural circuit that exists in the spinal cord of human beings [159]. Generally speaking, during a normal stable walking process, the basic motion control of muscles and legs is completed by the cerebellum, CPG, and spinal cord automatically. The subjective interference by human wills is not really necessary during such a process, unless there is something special happens.

For example, when people are walking on the street, they usually will not have to think about how their muscles should be powered or where their feet should locate for their next steps all the time. The control of the gaits is accomplished by the nervous system steadily and smoothly. The interference of human wills to gait control will only happen when something special, whether expected or unexpected, happens. Those special things can be the changes of the environment, for example, traffics, obstacles, directions, surfaces, and slopes of the road, or, the subjective changes of the walking rhythm by human beings.

For a steady walking process without those changes, which means the designated human being is walking straight and steadily on an ideal flat surface without disturbance, the corresponding human gait can be regarded as a cyclic process [124]. Figure 3.4 shows the phases of each gait cycle during such a straight and steady walking. The horizontal axis represents the gait time, while $T$ represents the time length of an entire gait cycle.

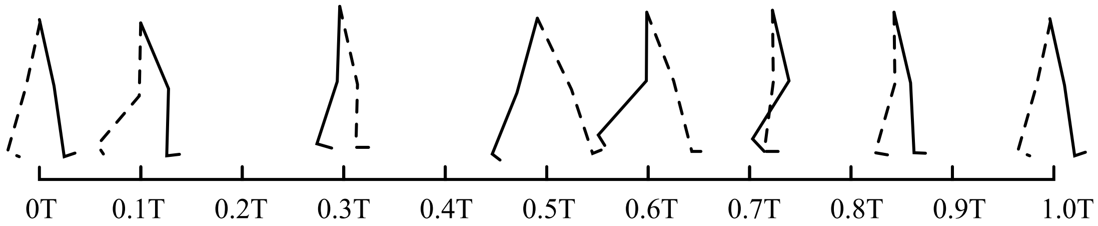

Figure 3.4: Normal Human Gait Phases.

The actual length of $T$ varies according to the physical conditions of each individual, but it usually will be a value of around 1.1 seconds [125].

An entire cycle of the gait can be divided into two phases. For each leg, there are a stance phase and a swing phase. In the stance phase, the foot part of the corresponding leg will touch the ground and bear the weight of the body, while the other leg is swinging forward. Then, in the swing phase, the original stance leg will become the swing leg. It will swing forward in the air while the weight of the body is now carried by the other leg. The swing phase ends when the swing leg touches the ground again. For the leg represented by the solid lines in Figure 3.4, its stance phase is the 0% to 60% part of an entire gait cycle, while its swing phase is the rest 60% to 100% part. The stance phase is longer than the swing phase because at the beginning and end of each stance phase, both feet will touch the ground and the body weight is shared by the two feet.

The status and phases of the lower limbs can be determined by monitoring the joint angle data in each gait cycle. Those joint angle data can also serve as the reference signal to help the controller of the exoskeleton robot to calculate a proper torque assistant scheme for the entire gait cycle. In real exoskeleton applications, many other data may also be applied to help to determine the status of the lower limbs, for example, the interactive forces, foot pressures, or EMG data, as previously stated in Chapter 2.1.

Since the motion control problem of the exoskeleton robot for walking assist missions under such a cyclic walking process is the main focus of this thesis, it is necessary to conduct a proper measurement for those relative human gait data in each cycle, in order to obtain better performance for the exoskeleton and provide a higher human-exoskeleton cooperation level.

### 3.3.2 Acquisition of Gait Data

The main gait angle data applied in this thesis is the joint angle data of the lower limbs. In order to obtain the designated joint angle data, a measurement system that contains a set of gyroscopes has been mounted on a test subject. A schematic diagram of the mounting method is shown in Figure 3.5.

The test subject is a 28-year-old male individual, with a height of 180 centimeters, a weight of 82 kilograms, and no obvious physical disease. Though only the gait data of one person has been collected in this section, it is worth noticing that the main purpose of this thesis is to study and verify the effectiveness of the proposed ILC algorithm, not to conclude the principles of human gait. The gait data collected from one person is enough to demonstrate the periodical characteristics of human gait cycles and to verify the effectiveness of the ILC algorithms dealing with such problems. Detailed studies about the principles of human gait have already been introduced in the previous subsection.

As shown in Figure 3.5, there are four gyroscopes in total mounted on the body of the test subject. One of the gyroscopes is mounted on the trunk of the test subject, while the other three are mounted on the thigh part, shank part, and the foot part of his left leg respectively. The red squares in the figure indicate the positions where the gyroscopes have been mounted, while the blue line segments are the fixing straps used to fasten the gyroscopes to the relevant body segments. These gyroscopes are connected to a laptop

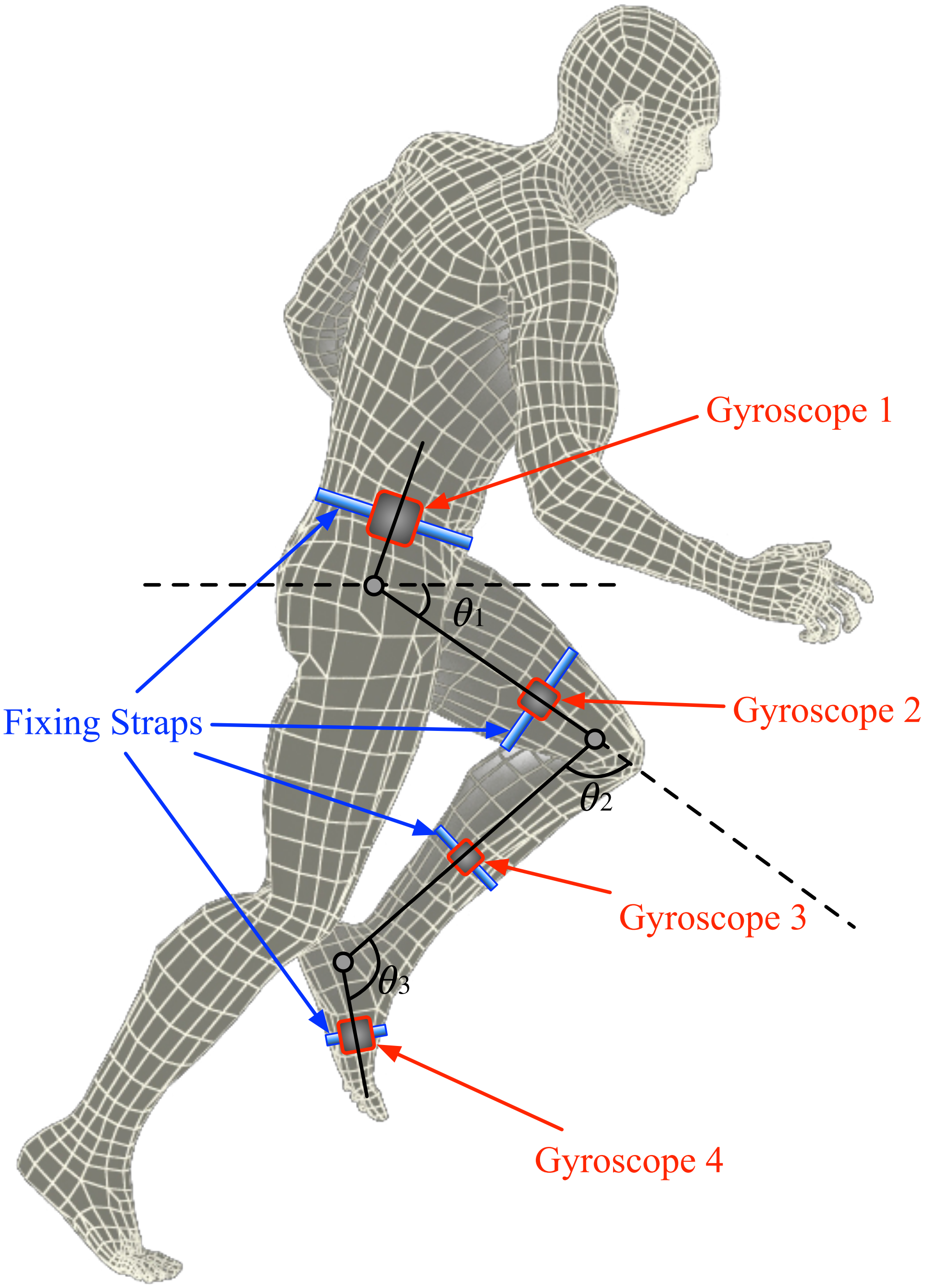

Figure 3.5: The Measurement Method of the Gait Angle Data.

computer via cables, while the data they obtain is also processed by the laptop.

The main data that have been measured in this case are three joint angles of the test subject during a straight walking process on a flat surface. The three joint angles include the hip angle, which is the angle between the thigh segment and the horizontal plane, marked as $\theta_1$ in the figure, the knee angle, which is the relative angle between the thigh segment and the shank segment, marked as $\theta_2$, and the ankle angle, which is the relative angle between the shank segment and the foot segment, marked as $\theta_3$. The gyroscopes can measure their relative angle data to the horizontal plane. By calculating the differences between the angle data of each body segment, the angle data of the three joints can be obtained. Meanwhile, only the data in the $X$-axis and $Y$-axis of Figure 3.2 has been considered. The data in the $Z$-axis which is vertical to the plane is ignored, as the motion scale of human lower limbs in this direction during normal walking is minor, and the exoskeleton model studied in this thesis has no powered actuators on this axis.

The straight walking process needed for the measurement is carried out in a long empty corridor of a building. The walking process has been done three times and a less noise set of joint data has been picked among them. The raw data is filtered via the median average filtering method. The results are demonstrated in the following figures. Figure 3.6 shows the angle variation of the hip joint in a typical gait cycle, while Figure 3.7 shows the angle variation of the knee joint. The angle variation of the ankle joint is shown in Figure 3.8. These data are selected from a series of gait cycles as they present the typical variation of the joint angles during the walking process smoothly. It can be verified that these measured gait data fit the common shapes, amplitude limits, and time length of normal human gait cycle data obtained in other research papers [125, 160]. Hence, these measured data can represent the status of typical human gaits. Also, these gait data for single gait cycles will serve as the angle reference signals for the simulation of the ILC algorithms in the following chapters.

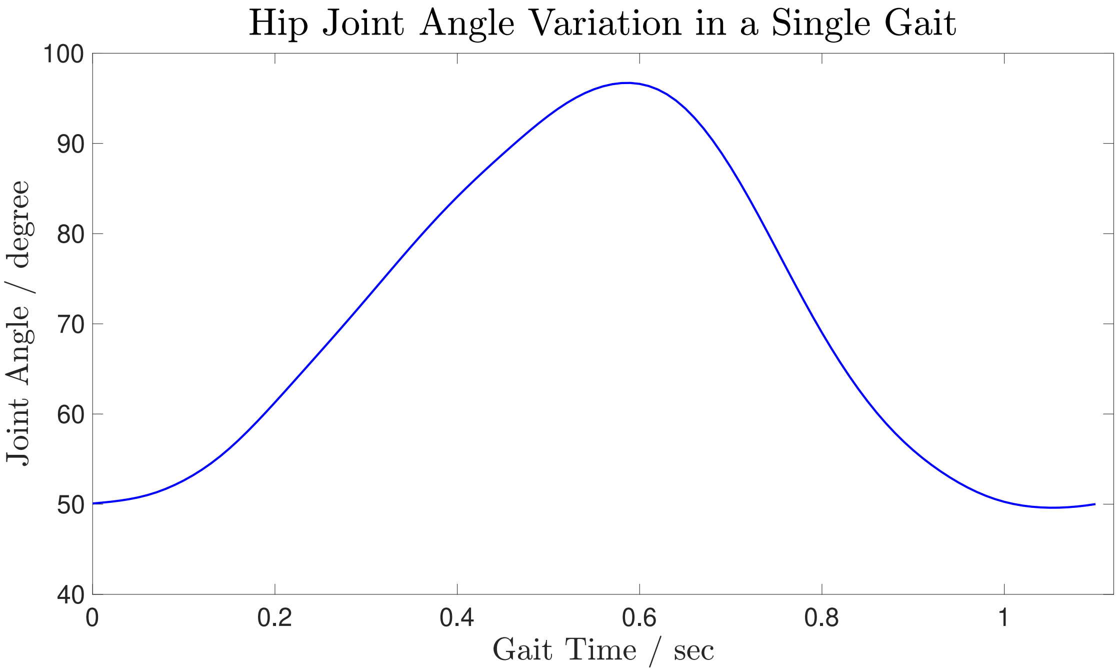

Figure 3.6: Variation of the Hip Joint Angle during a Single Gait.

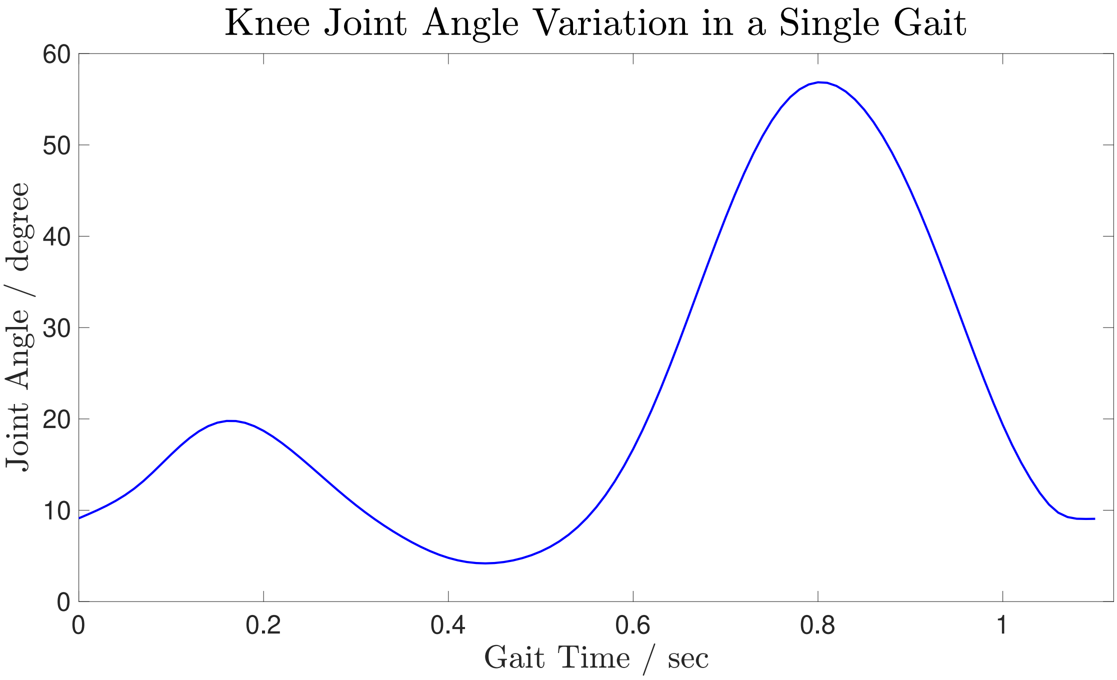

Figure 3.7: Variation of the Knee Joint Angle during a Single Gait.

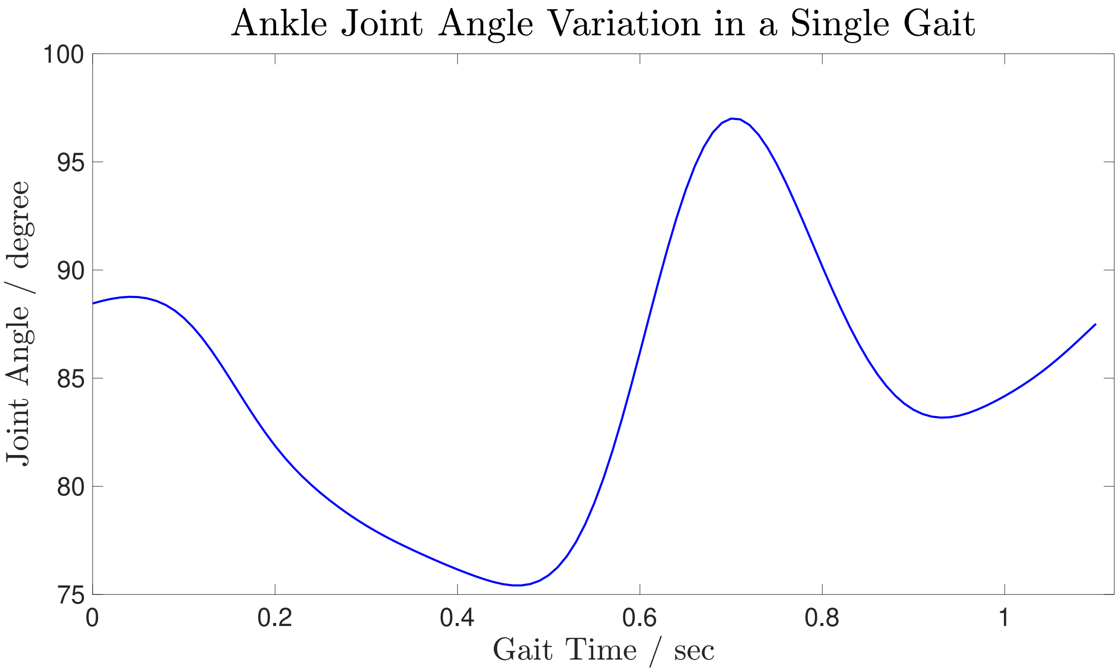

Figure 3.8: Variation of the Ankle Joint Angle during a Single Gait.

Other than the single gait cycle data, two series of joint angle data in continuous gait cycles have also been collected for further tests and comparisons. These continuous gait cycle data are measured when the test subject is walking at a slightly faster speed. The corresponding angle data are shown in Figure 3.9 and 3.10, where the blue line is the continuous knee joint angle and the red line is the continuous hip joint angle. The continuous gait data serve as the reference signals for adaptive oscillator method cases as a comparison to the ILC methods.

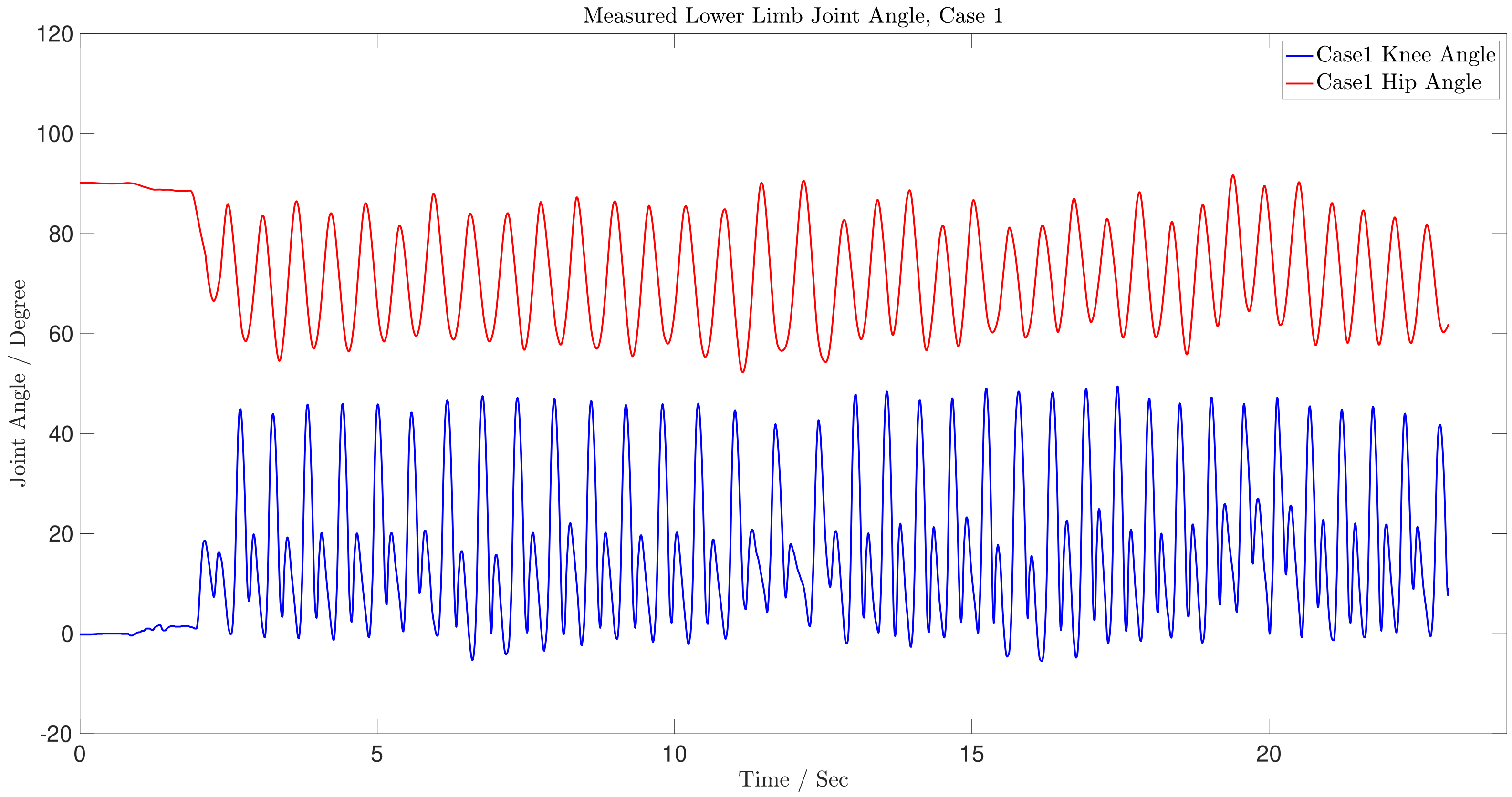

Figure 3.9: Measured Lower Limb Joint Angle, Case 1

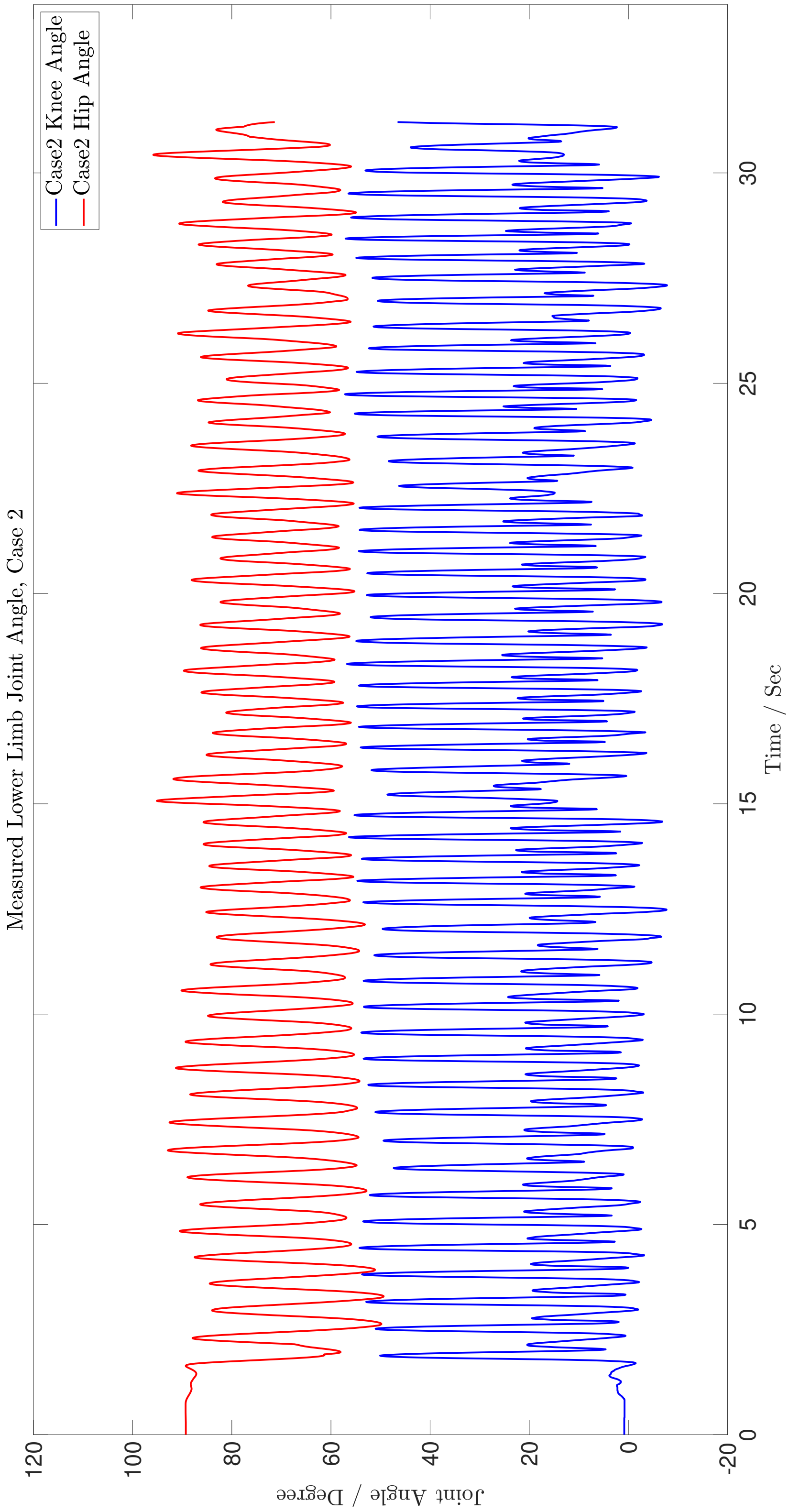

Figure 3.10: Measured Lower Limb Joint Angle, Case 2

## 3.4 Conclusion

This chapter briefly introduced the basic concepts and preliminaries related to the ILC algorithms, robotics, and gait data measurements. Firstly, the background of ILC methods, typical ILC algorithms for linear and nonlinear systems, and relative designing criteria are given. Then, the kinematic and dynamic model of the robot, as well as the dynamic model for the combination of human lower limbs and the exoskeleton robot, are discussed, followed by a simple description for the biomechanics of the human gaits. Finally, the joint angle data for both single gait cycles and continuous gait cycles are given as well as their measurement methods.

# Chapter 4

# Iterative Learning Control for the Shank Part of Lower Limb Exoskeleton

A novel ILC algorithm, which combines the feedback linearization method with the original ILC algorithms, has been proposed in this chapter. Furthermore, the proposed algorithm has been applied to a lower limb exoskeleton robot model to verify its effectiveness. The proposed algorithm can also combine with the NOILC and Dual Internal Model Structure (DIMS) ILC algorithms to improve their performances. The relative simulation results, as well as the comparison between the proposed ILC algorithm and the original ILC algorithms, have also been given in this chapter. Section 4.1 introduces the original ILC algorithm and Section 4.2 the target exoskeleton robot. Section 4.3 presents the simulation results of the original ILC algorithm while Section 4.4 introduces the proposed novel ILC algorithm to deal with the problems that exist in the original ILC algorithm. Section 4.5 presents the combination of the proposed algorithm and the NOILC method. Section 4.6 shows its application with the DIMS ILC algorithm. Finally, section 4.7 concludes the chapter.

## 4.1 Original Iterative Learning Control Algorithm

For a fixed and finite time interval $t \in [0, T]$, consider the following continuous time-invariant dynamic system given by:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{4.1}$$

where $x(t) \in R^n$ is the state vector, $y(t) \in R^m$ is the output vector, $u(t) \in R^p$ is the control input vector, $f(\cdot) \in R^n$ are smooth vector-valued function, and $B, C$ are constant state matrices of proper dimensions. The reference signal of the system in time interval $t \in [0, T]$ is given by $r(t)$.

The repeat time of the control operation period is called iteration trials. For the $k$-th iteration trial, the control updating law of the ILC algorithm is given by:

$$u_{k+1}(t) = u_k(t) + \gamma \dot{e}_k(t), \tag{4.2}$$

where $\gamma \in R^{p \times m}$ is a constant learning gain matrix, and $e_k(t) \in R^m$ is the tracking error of the system in the $k$-th iteration trial given by:

$$e_k(t) = r(t) - y_k(t). \tag{4.3}$$

According to [82], define a vector norm for $e(t)$, which is also a vector-valued function, on the fixed finite time interval $t \in [0, T]$ as:

$$||e(t)||_j = \sup_{0 \leq t \leq T} \{\exp(-jt)||e(t)||_\infty\}. \tag{4.4}$$

If for the entire time interval $t \in [0, T]$, the following condition holds:

$$\begin{cases} y_k(0) = r(0), \\ ||I_m - CB\gamma||_\infty < 1. \end{cases} \tag{4.5}$$

where $I_m$ is an $m$-dimensional identity matrix, then there exists constant $j \geq 0$ and $0 \leq q \leq 1$ such that:

$$||\dot{e}_{k+1}(t)||_j \leq q||\dot{e}_k(t)||_j, \tag{4.6}$$

holds for any $k$. The corresponding proof has been demonstrated in [129]. Hence by choosing a proper learning gain matrix $\gamma$ according to equation (4.5), there is:

$$\lim_{k \to \infty} ||\dot{e}_k(t)||_j = 0, t \in [0, T]. \tag{4.7}$$

Substituting equation (4.3) into (4.7) leads to:

$$\lim_{k \to \infty} \frac{\mathrm{d}}{\mathrm{d}t} (r(t) - y_k(t)) = 0, t \in [0, T]. \tag{4.8}$$

Considering the initial condition $y_k(0) = r(0)$ in equation (4.5) which means $e_k(0) = 0$, equation (4.8) implies that:

$$\lim_{k \to \infty} y_k(t) = r(t). \tag{4.9}$$

When the iteration number increases, the system output $y_k(t)$ will gradually approach to the reference $r(t)$ and the error between them will go to zero as iteration number goes to infinity. Hence, the tracking problem of system (4.1) can be solved by the ILC updating law (4.2).

## 4.2 Dynamic for the Shank Part of the Lower Limb Exoskeleton

As stated in the previous section, in the rehabilitation applications of exoskeleton robot, the exoskeleton usually moves along a predefined repetitive gait trajectory. It makes the ILC method suitable to be applied to such conditions. To verify the effectiveness of the proposed algorithm, consider the rigid body model of lower limb exoskeleton as shown in Figure 4.1. At this stage, only the one DoF shank part has been considered, but the algorithm can be extended to the full lower limb.

As shown in Figure 4.1, the shank part is modeled as a uniform rigid body stick from knee joint point $K$ to ankle joint point $A$. The stick has a length of $l$ and a mass of $m_s$, which is the sum of mass for both the human shank part and the exoskeleton shank part. A particle with the mass of foot part $m_f$ is attached to ankle point $A$. $\theta$ is the angle between the shank part and vertical direction. An actuator is mounted at the knee position of the exoskeleton, which gives a torque of $\tau$ to the shank part and drives the entire movement of the system.

According to the equation of motion for the system, there is:

$$M\ddot{\theta} + f(\dot{\theta}, \theta) + g(\theta) = \tau, \tag{4.10}$$

where $M$ represents the inertia of shank, $f(\dot{\theta}, \theta)$ represents the effects of centrifugal force and Coriolis force, and $g(\theta)$ represents the torque to the knee joint $K$ by gravity [124].

Considering the shank part only, the directions of the centrifugal force and Coriolis force are along the stick, pointing at the far side of knee joint $K$, which means they will not generate torque to the knee joint. After calculating the inertia and gravity, the dynamic

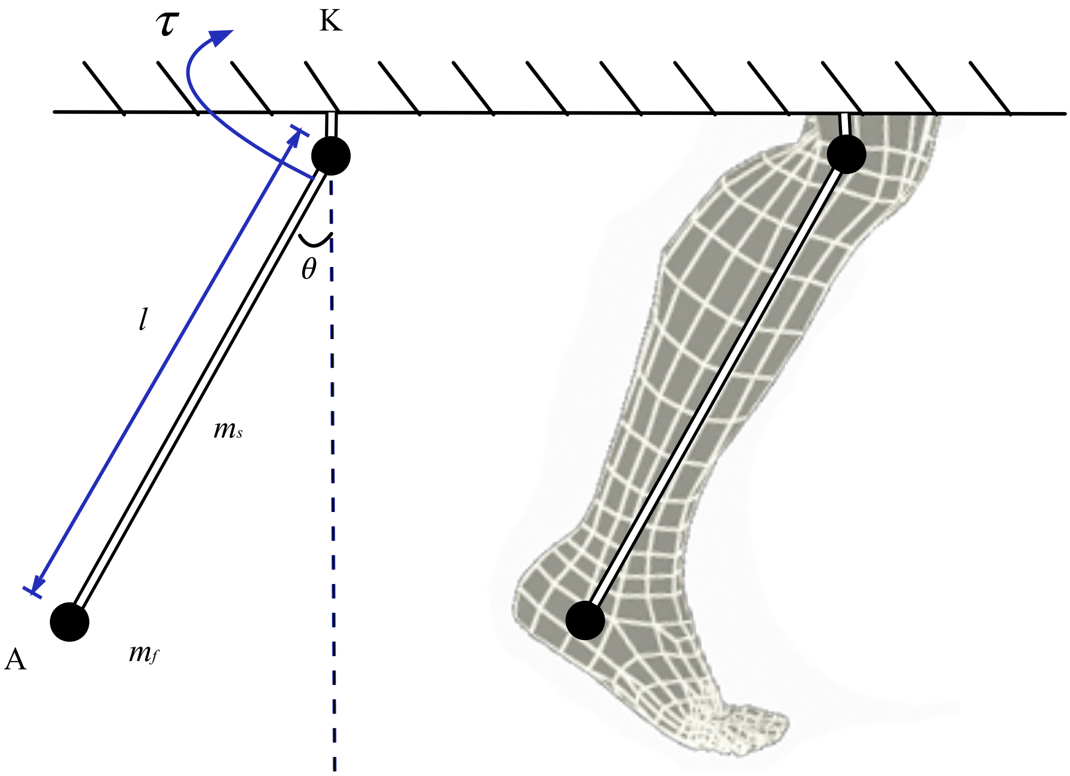

Figure 4.1: Rigid Body 1-DoF Shank Model.

model of the shank part can be described by:

$$c_1\ddot{\theta} + c_2 \sin\theta = \tau, \tag{4.11}$$

where $c_1$ and $c_2$ are real constants given by:

$$\begin{aligned} c_1 &= \frac{m_s l^2}{3} + m_f l^2, \\ c_2 &= \left(\frac{m_s}{2} + m_f\right) l g_0, \end{aligned} \tag{4.12}$$

$g_0$ in equation (4.12) is the gravitational acceleration on earth which equals to $9.8\mathrm{m/s^2}$.

Set the torque generated by knee joint actuator as the system input $u$, which gives $u = \tau$, and define the state vector $x$ by:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}. \tag{4.13}$$

The dynamic of the shank part in time interval $t \in [0, T]$ can be described by:

$$\begin{aligned} \dot{x_1}(t) &= x_2(t), \\ \dot{x_2}(t) &= -\frac{c_2}{c_1}\sin x_1(t) + \frac{1}{c_1}u(t). \end{aligned} \tag{4.14}$$

Define the shank angle $\theta$ as the output of the system, then the output vector is:

$$y = Cx, \tag{4.15}$$

where $C = [1, 0]$.

## 4.3 Simulation Results of the Orignal ILC Algorithm

According to [125, 161] and Table 3.1, the length of shank is approximately $28.5\%$ of body height, while its mass is $4.65\%$ of body weight. The mass of foot is $1.45\%$ of body weight. Assuming the wearer of the exoskeleton is of $1.80$m height and $75$kg weight, the weight of exoskeleton shank part itself is $1.5$kg, then the specification parameters in the shank model can be obtained by $l = 0.51$m, $m_s = 4.99$kg and $m_f = 1.09$kg, which lead to the value of the constants given by $c_1 = 0.72$ and $c_2 = 17.92$.

Then the dynamic model of the shank part in time interval $t \in [0, T]$ can be written as:

$$\begin{aligned} \dot{x_1}(t) &= x_2(t), \\ \dot{x_2}(t) &= -24.89 \sin x_1(t) + 1.39u(t), \\ y(t) &= x_1(t). \end{aligned} \tag{4.16}$$

The reference signal of the system is the target trajectory of the shank angle in time interval $t \in [0, T]$, as shown in Figure 4.2. It represents the shank angle variation during a walking gait cycle, which is calculated via the position data of human lower limb joints in [125], captured by cameras with a sampling time of $0.15$ second. The average gait cycle time is $T = 1.1$ seconds.

Since the wearer can choose to start at any reachable shank angle, and the reference signal can be known in prior as well, it is reasonable to assume that the wearer can start the iteration at the same initial angle with the reference, which gives:

$$y(0) = r(0). \tag{4.17}$$

The goal of control system design is to generate a proper control input $u$ so that the trajectory of the shank angle can track the proposed reference.

Applying the ILC updating law (4.2) to the dynamic system (4.16), for learning gain $\gamma = 0.5$, the system output in iteration 1 to 4 are shown in Figure 4.3, while iteration 30 to 33 are shown in Figure 4.4 as the error has converged. The corresponding tracking error are shown in Figure 4.5 and 4.6. It can be observed that, as the iteration trial number increases, the system output gradually approaches the reference trajectory.

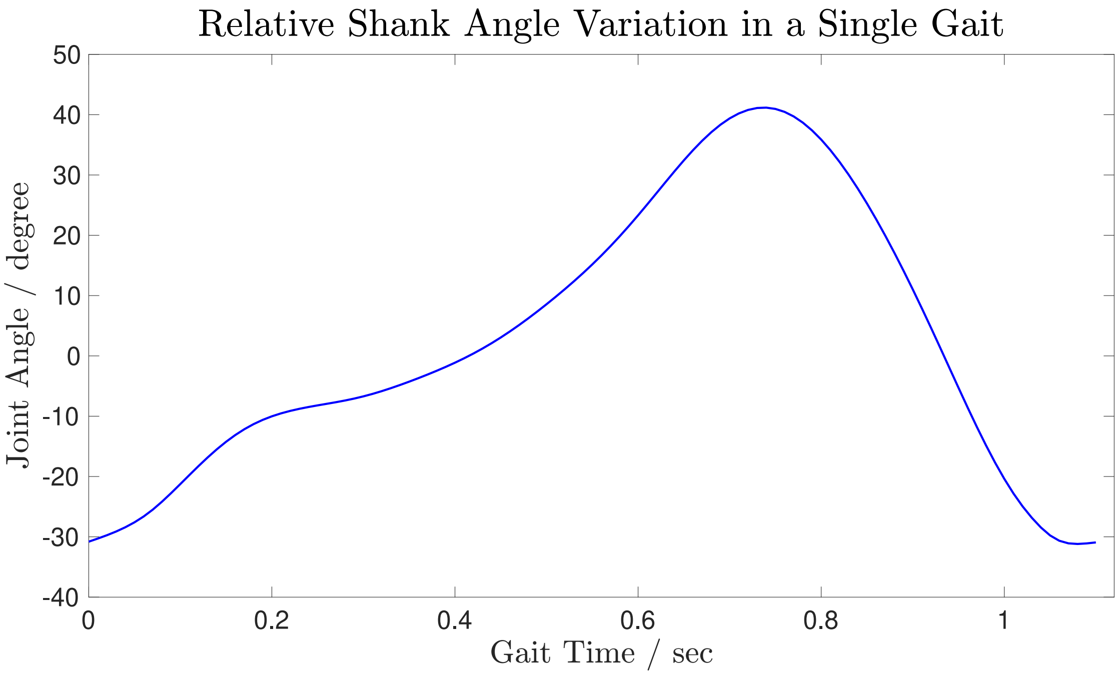

Figure 4.2: The Reference of Shank Angle in a Single Gait Cycle.

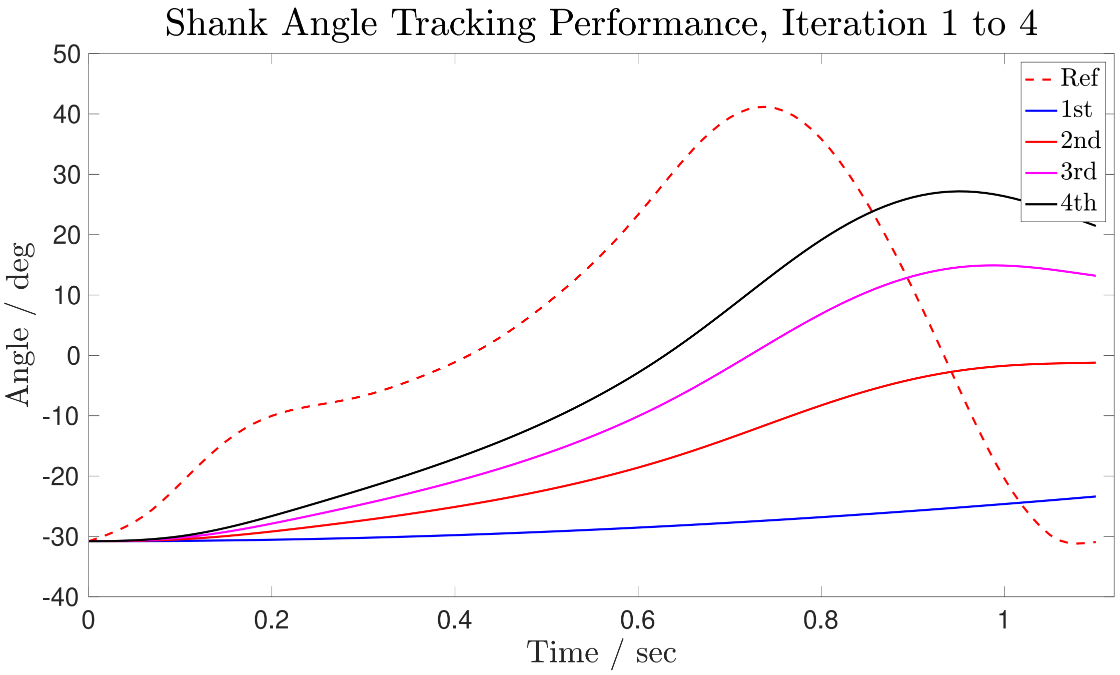

Figure 4.3: Angle Tracking Performance, D-Type ILC, Iteration 1 to 4.

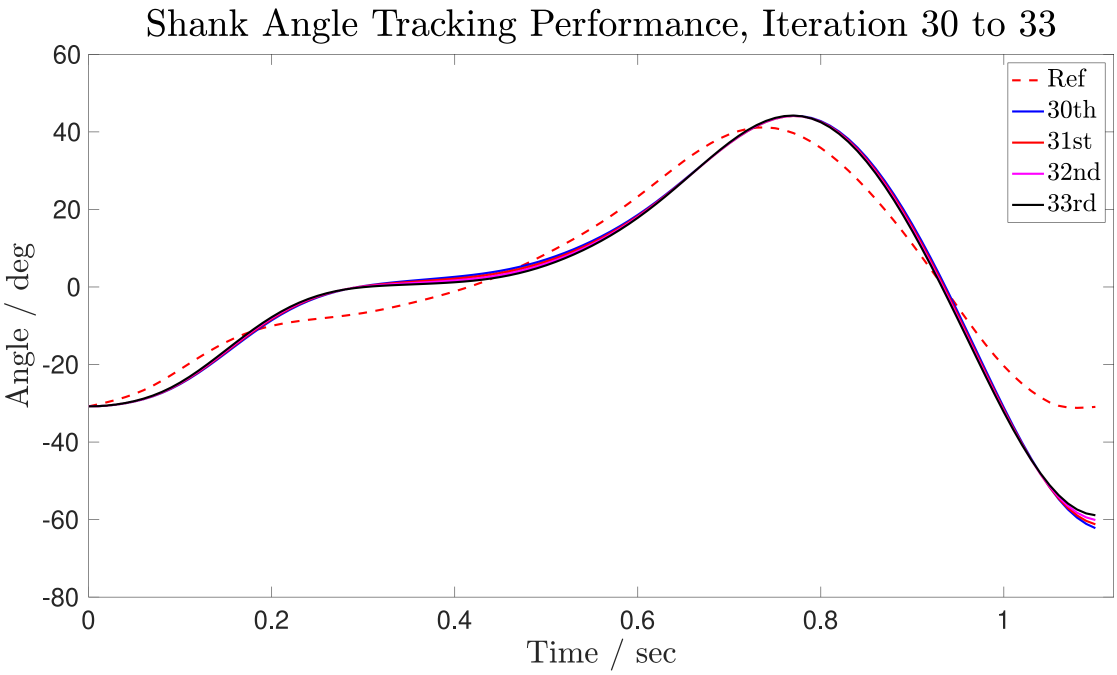

Figure 4.4: Angle Tracking Performance, D-Type ILC, Iteration 30 to 33.

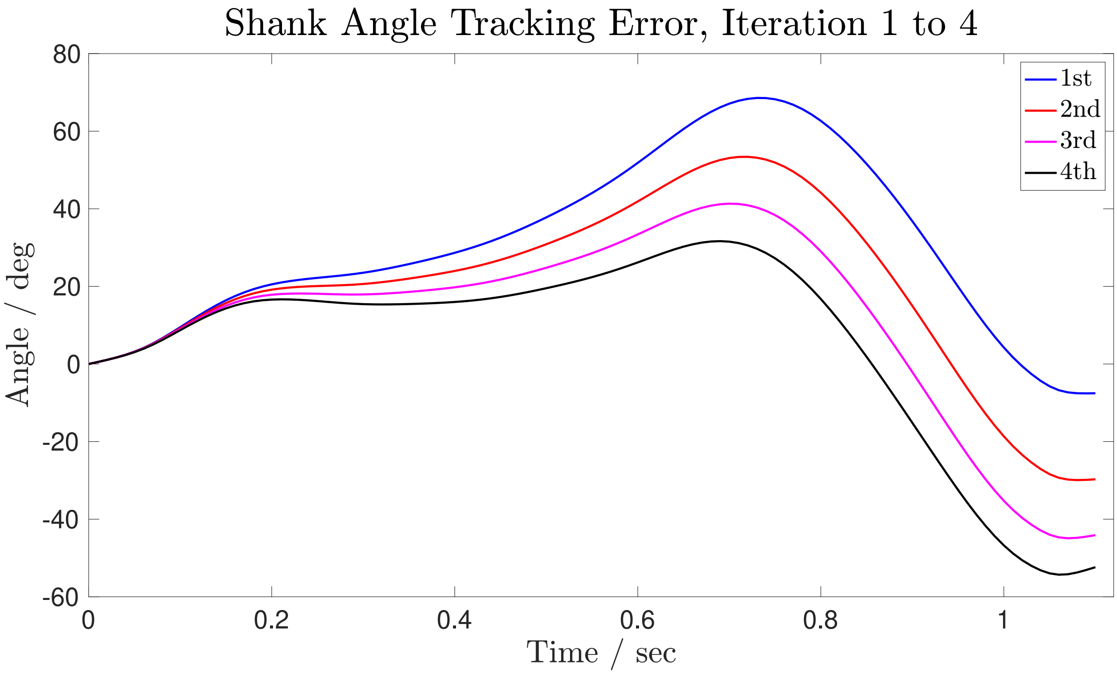

Figure 4.5: Angle Tracking Error, D-Type ILC, Iteration 1 to 4.

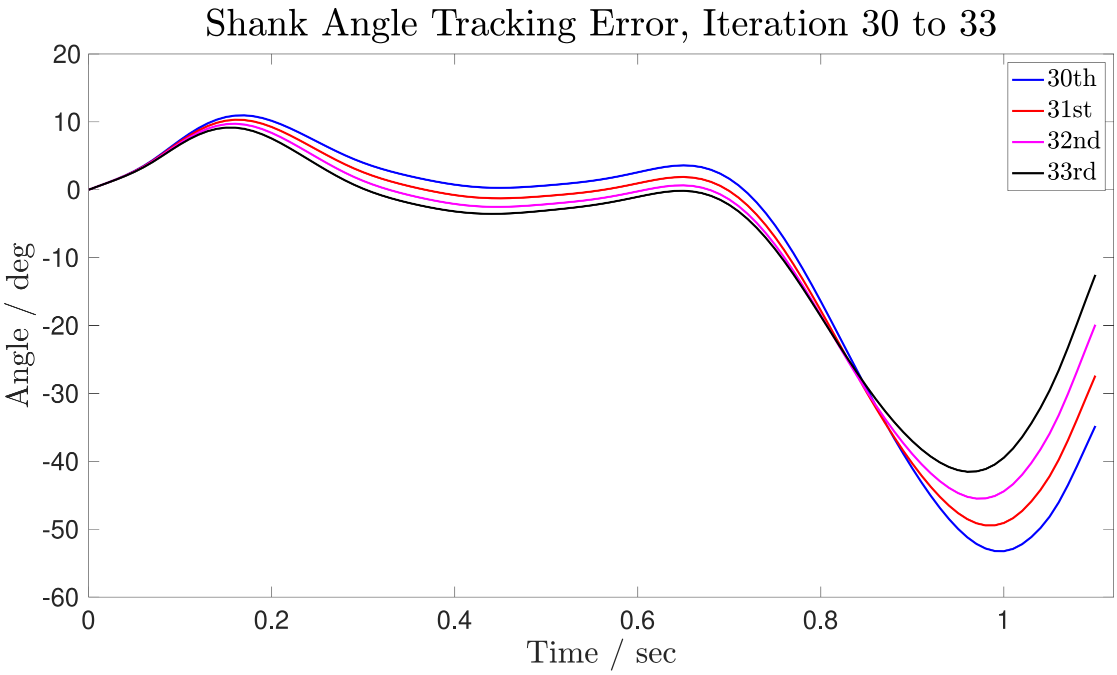

Figure 4.6: Angle Tracking Error, D-Type ILC, Iteration 30 to 33.

Figure 4.7 and 4.8 indicate the system tracking performance in terms of the average absolute error for each iteration trial, with different learning gain values. The choice of the learning gain $\gamma$ will affect the convergence speed of error. For $\gamma$ value between $1.0$ and $2.0$, the system error has its highest convergence rate. Its amplitude decreases in a higher rate in the first 10 iteration trials but slower in the following trials.

For dynamic system (4.16), condition (4.5) holds for any $\gamma$ value, which means theoretically the tracking error is guaranteed to converge for any $\gamma$. However, for $\gamma$ beyond $2.5$, the tracking error will increase dramatically at the beginning, which leads to a worse tracking performance for the entire system compared to the smaller $\gamma$ values. Such a result is demonstrated in Figure 4.8.

Although the original ILC updating law can solve the tracking problem for the system (4.16), the convergence speed of the error is far from satisfaction. As presented in Figure 4.4 and 4.6, even after 30 iteration trials, there still exist obvious errors between

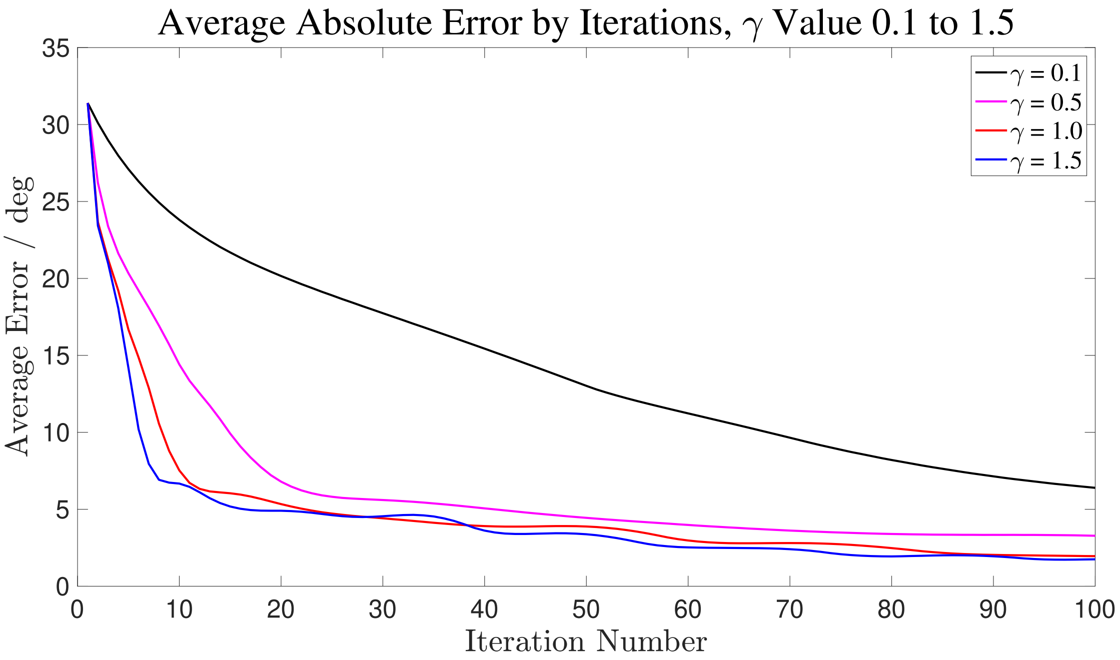

Figure 4.7: Average Absolute Error, D-Type ILC, Small $\gamma$ Values.

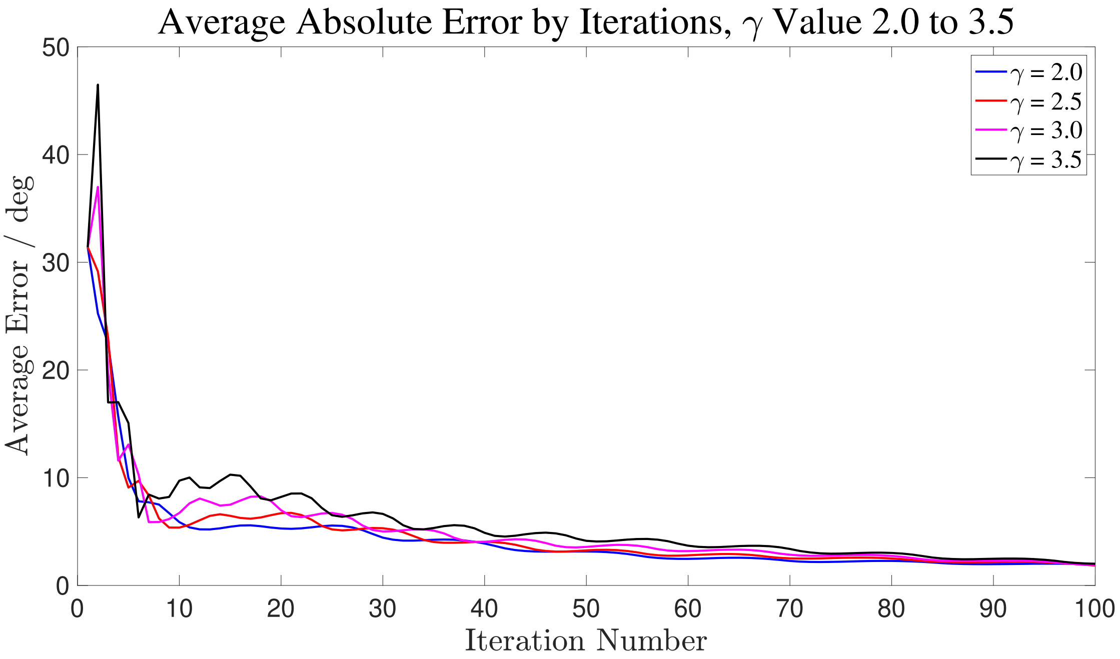

Figure 4.8: Average Absolute Error, D-Type ILC, Large $\gamma$ Values.

the reference and the system output. To solve this problem, an improved controller is introduced in the following section.

## 4.4 ILC with Feedback Linearization

As shown in Figure 4.2, the reference signal is bounded in a neighbour of zero for time interval $t \in [0, T]$. However, the open-loop system (4.16) is an unstable system with nonlinearity. The nature divergence trend of its open-loop response has decreased the control performance of the original ILC updating law (4.2) since a part of its abilities has been used to stabilize the system. Hence, by implementing a feedback linearization structure, the open-loop nonlinear system can be converted to a linear one with desired pole positions [162] and the entire control performance can be improved.

Taking the derivatives of $y$ in system (4.16), for $t \in [0, T]$, there is:

$$\begin{aligned} \dot{y}(t) &= x_2(t), \\ \ddot{y}(t) &= -24.89 \sin x_1(t) + 1.39u(t). \end{aligned} \tag{4.18}$$

Define a new feedback variable $v(t)$ by:

$$v(t) = -24.89 \sin x_1(t) + 1.39u(t), \tag{4.19}$$

the linearized system of (4.16) is now given by:

$$\begin{aligned} \dot{x_1}(t) &= x_2(t), \\ \dot{x_2}(t) &= v(t). \end{aligned} \tag{4.20}$$

The state feedback law for the linearized system (4.20) can then be designed as:

$$v(t) = -a_1 x_1(t) - a_2 x_2(t), \tag{4.21}$$

which stabilizes the system with constants $a_1 > 0$ and $a_2 > 0$. Define $w(t)$ as the new ILC updating law of the linearized system in $k$-th iteration trial, which equals to:

$$w_{k+1}(t) = w_k(t) + \gamma \dot{e}_k(t). \tag{4.22}$$

The reference tracking form of the linearized feedback law (4.21) can be rewrite as:

$$v(t) = -a_1 \left(x_1(t) - w(t)\right) - a_2 x_2(t). \tag{4.23}$$

Combining (4.19) with (4.23), there is:

$$-24.89 \sin x_1(t) + 1.39 u(t) = -a_1 \left(x_1(t) - w(t)\right) - a_2 x_2(t). \tag{4.24}$$

Then, the control input of the original system (4.16) can be given by:

$$u(t) = \frac{1}{1.39} \left(24.89 \sin x_1(t) - a_1(x_1(t) - w(t)) - a_2 x_2(t)\right). \tag{4.25}$$

With the new proposed ILC algorithm with Feedback Linearization (FL), for $a_1 = 25$, $a_2 = 10$, learning gain $\gamma = 0.5$, the system output in iteration 1 to 4 and iteration 10 to 13 are shown in Figure 4.9 and 4.10 respectively. The corresponding tracking error are shown in Figure 4.11 and 4.12.

Compared to the results in the original ILC algorithm case, the tracking performance has been improved dramatically. Within only 4 iteration trials, the tracking error has already decreased to a small value.

The variation of the average absolute error by iteration numbers for different learning gain values are shown in Figure 4.13 and 4.14. For learning gain $\gamma$ between $0.3$ and $0.8$, the tracking error has its highest convergence rate, where its amplitude can decrease to almost $0$ in less than 10 iteration trials.

For $\gamma$ larger than $1.2$, there is a similar phenomenon to the original ILC algorithm case, where the updating law will generate a huge overshoot in the beginning and reduce the control performance for the entire system, as shown in Figure 4.14.

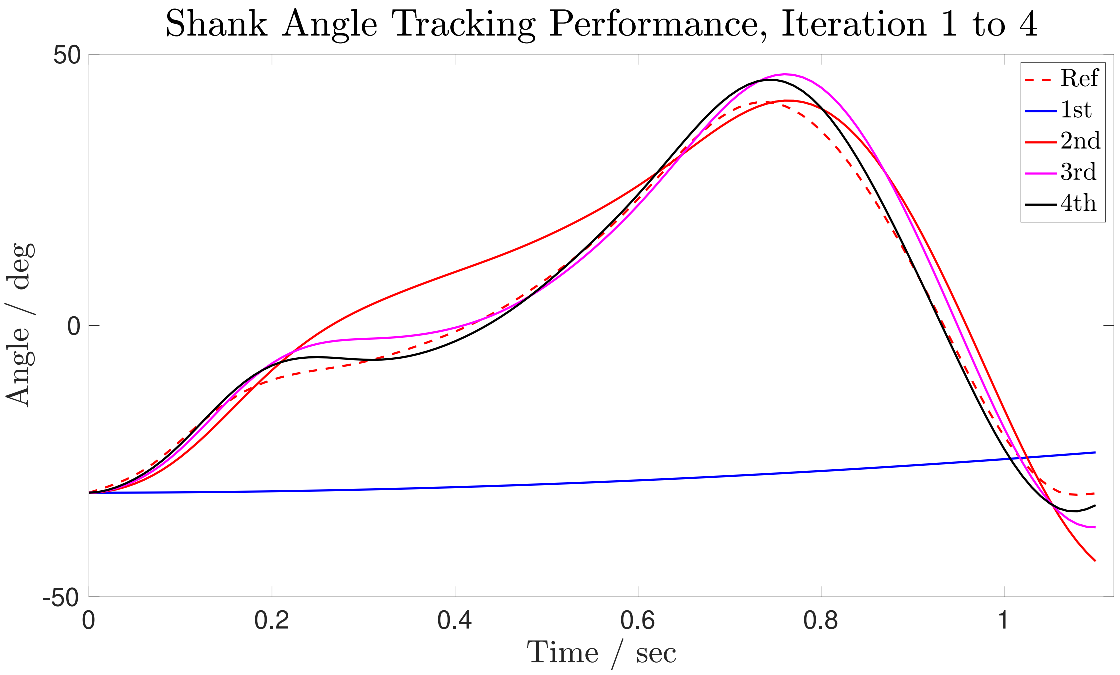

Figure 4.9: Angle Tracking Performance, ILC with FL, Iteration 1 to 4.

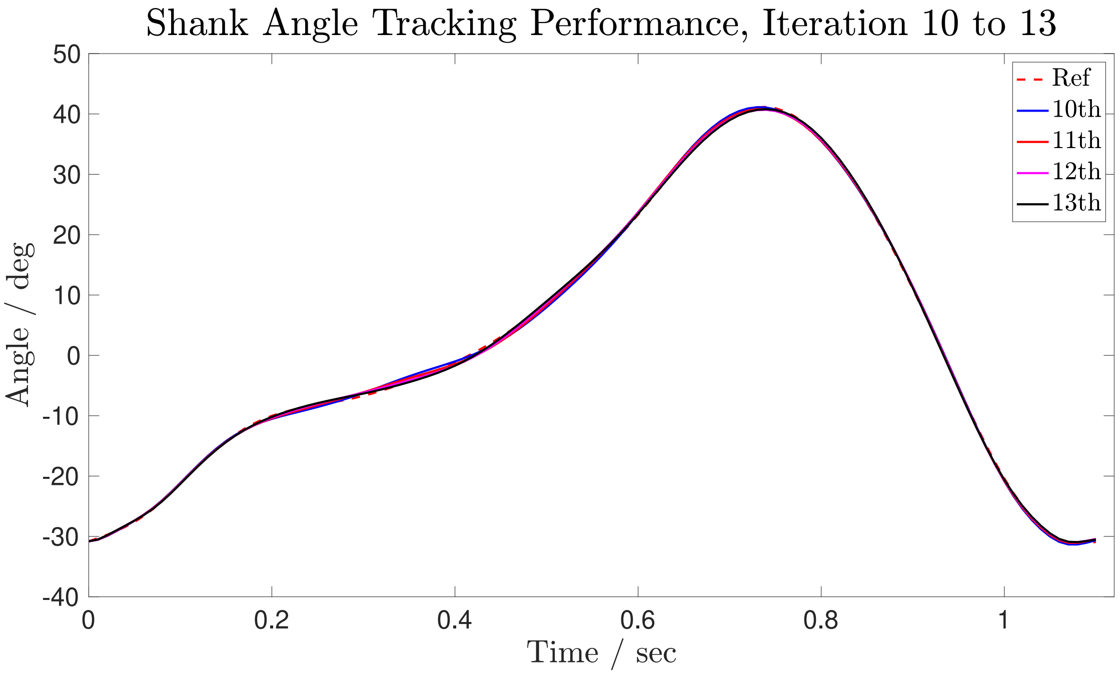

Figure 4.10: Angle Tracking Performance, ILC with FL, Iteration 10 to 13.

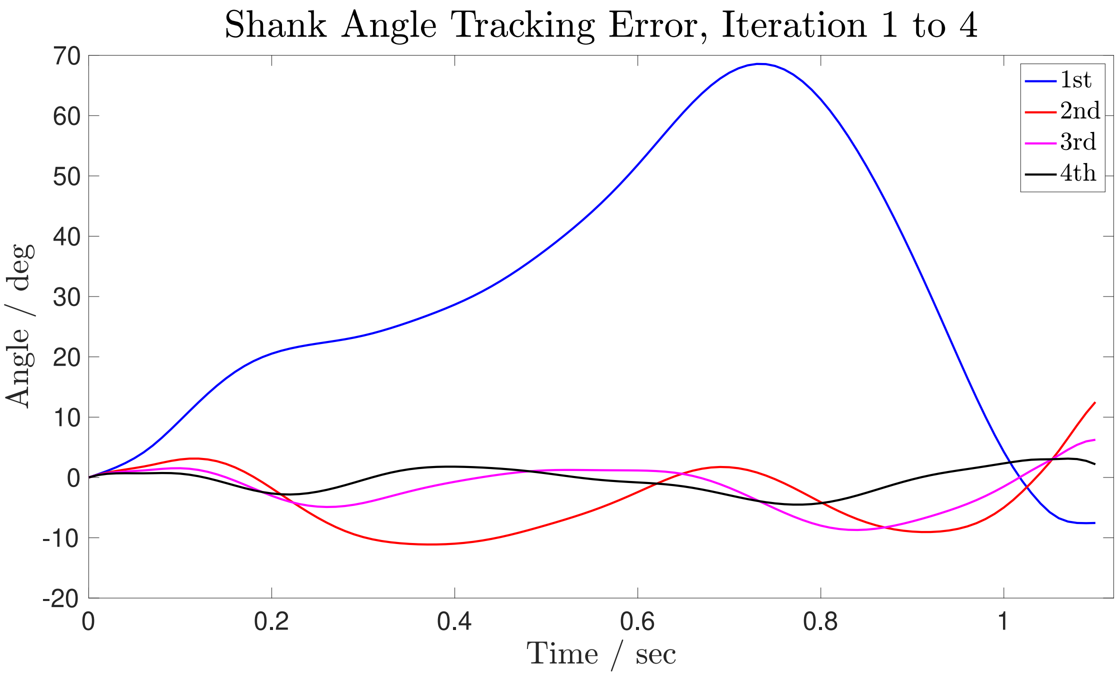

Figure 4.11: Angle Tracking Error, ILC with FL, Iteration 1 to 4.

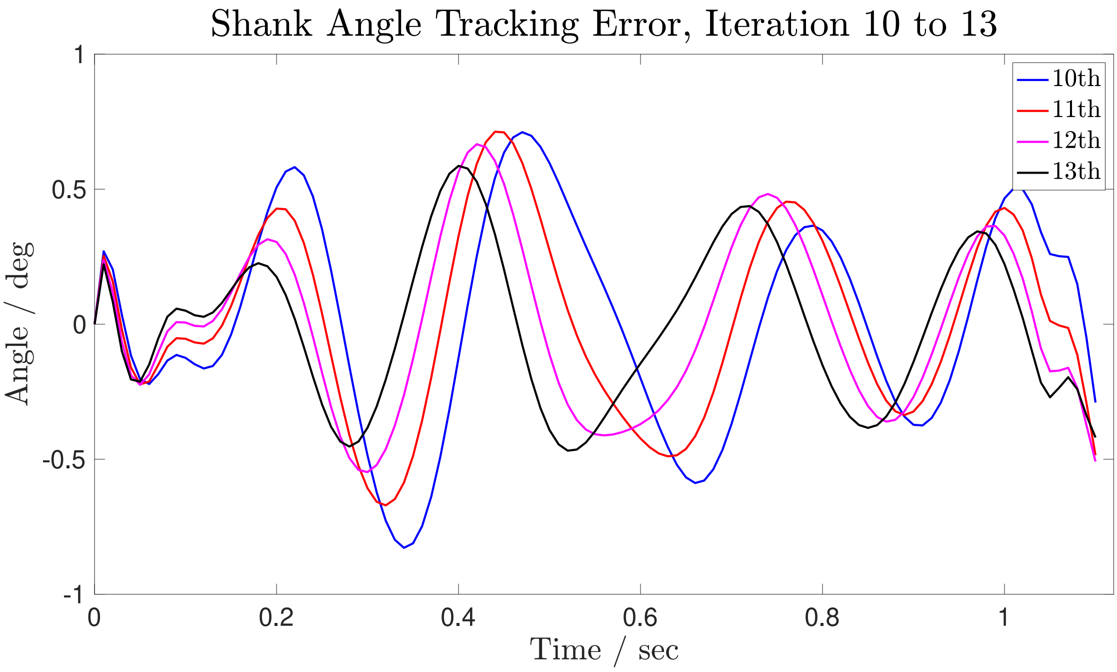

Figure 4.12: Angle Tracking Error, ILC with FL, Iteration 10 to 13.

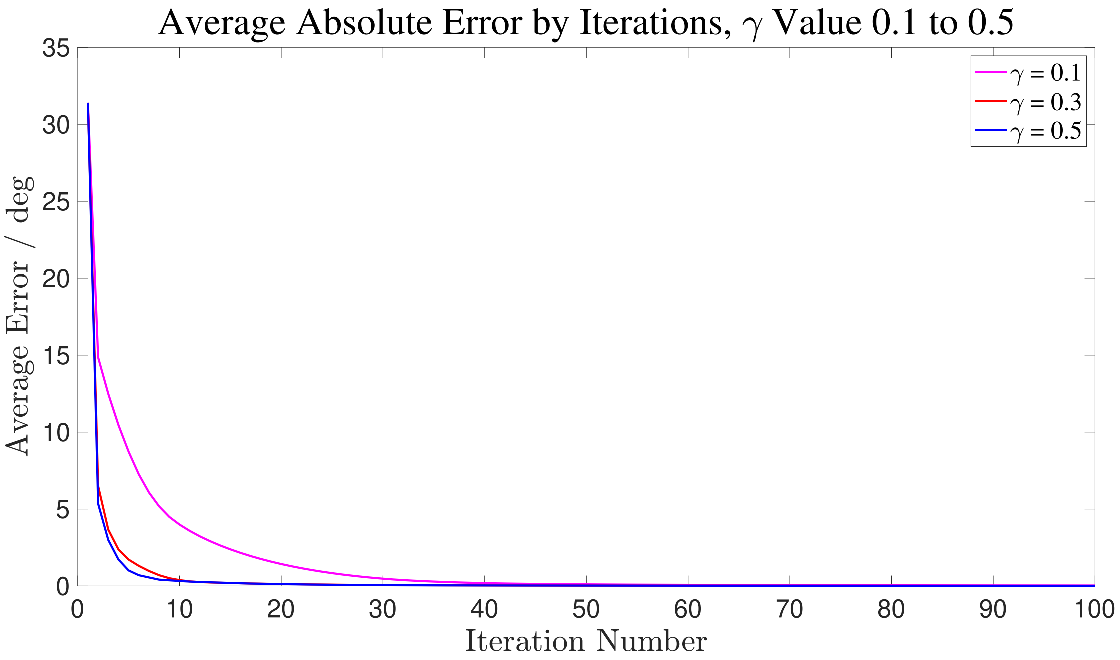

Figure 4.13: Average Absolute Error, ILC with FL, Small $\gamma$ Values.

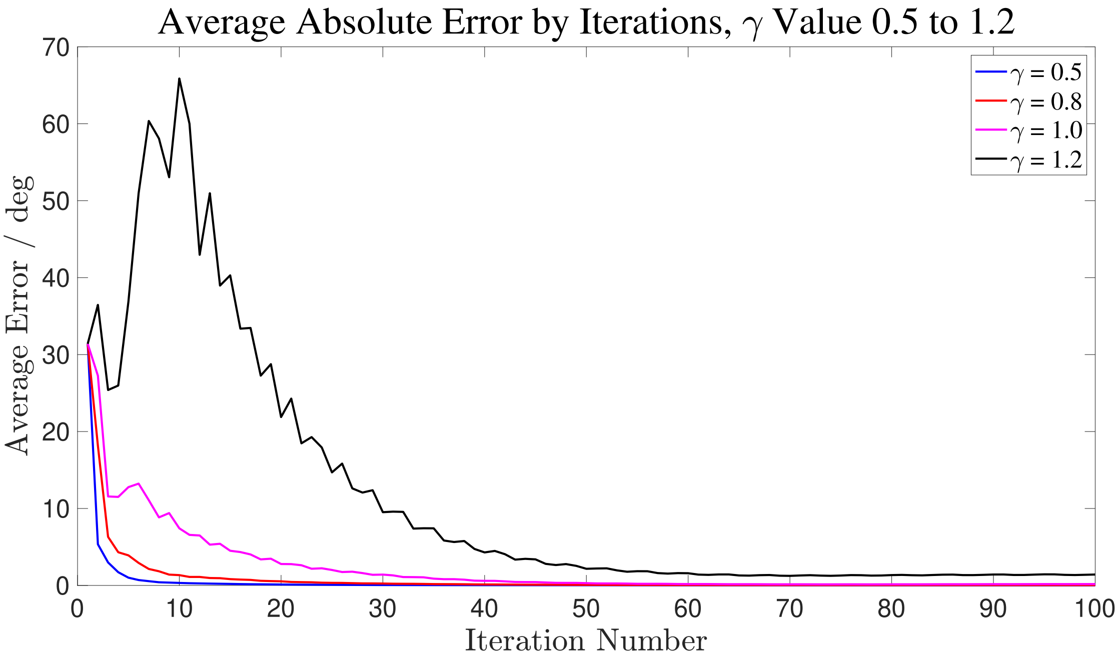

Figure 4.14: Average Absolute Error, ILC with FL, Large $\gamma$ Values.

Finally, for any sensor applied in the practical system, the measurement noise will always exist. By adding Gaussian noise with an average value of 0 and variance of 6.25 to $y(t)$ in system (4.16), the control performance with about $5\%$ measurement noise is simulated. Although facing an obvious degradation, the ILC algorithm with feedback linearization can still work in such a scenario, as shown in Figure 4.16, while its convergence speed and average error are superior to normal ILC method as well, as shown in Figure 4.15. The comparison for the error convergence performances are shown in Figure 4.17. However, as presented in these figures, due to the basic principle of the ILC method, it is not good at handling non-repeating random noises. To guarantee the smooth operation of the robot, an additional filter for the measured data will be required.

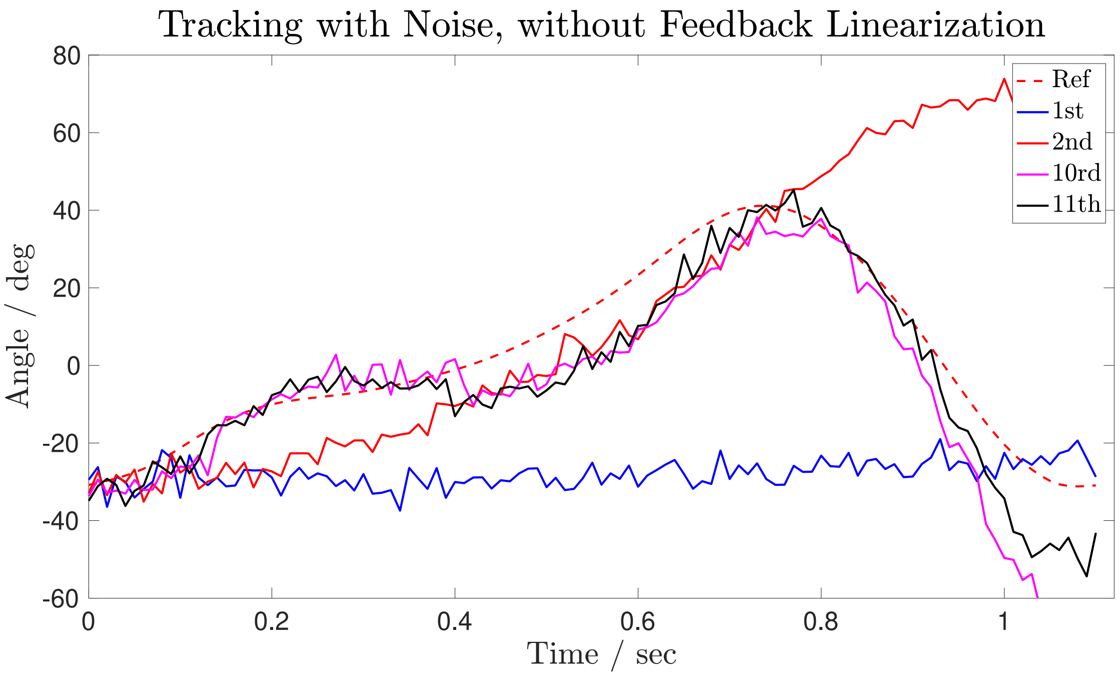

Figure 4.15: Tracking Performance with Noise, D-Type ILC.

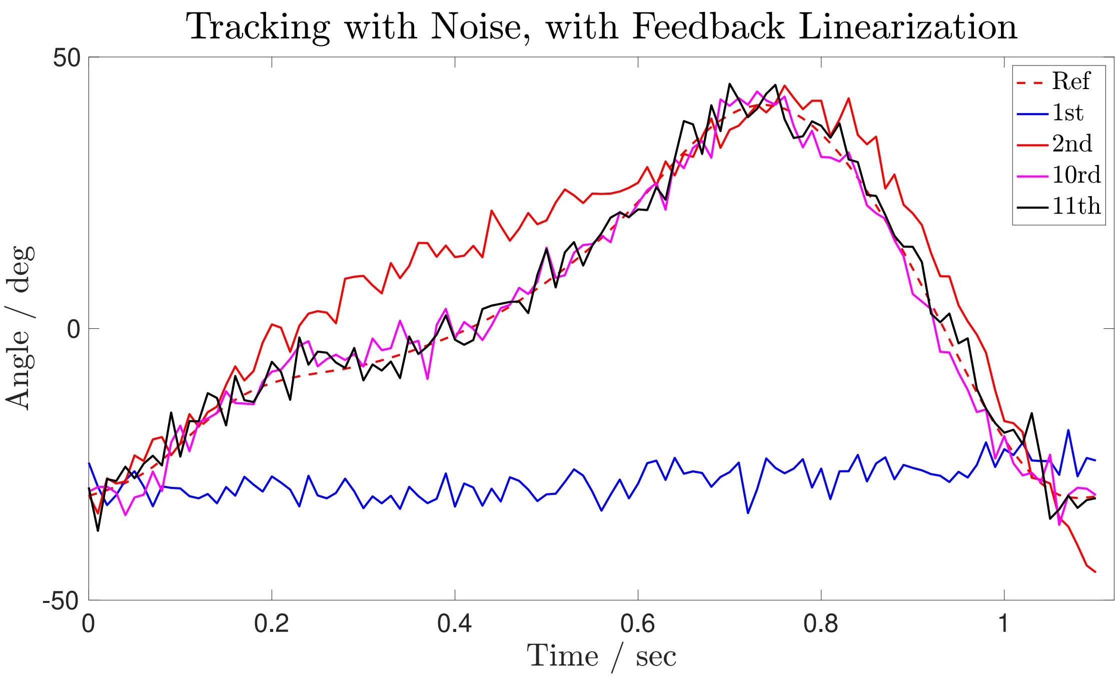

Figure 4.16: Tracking Performance with Noise, ILC with FL.

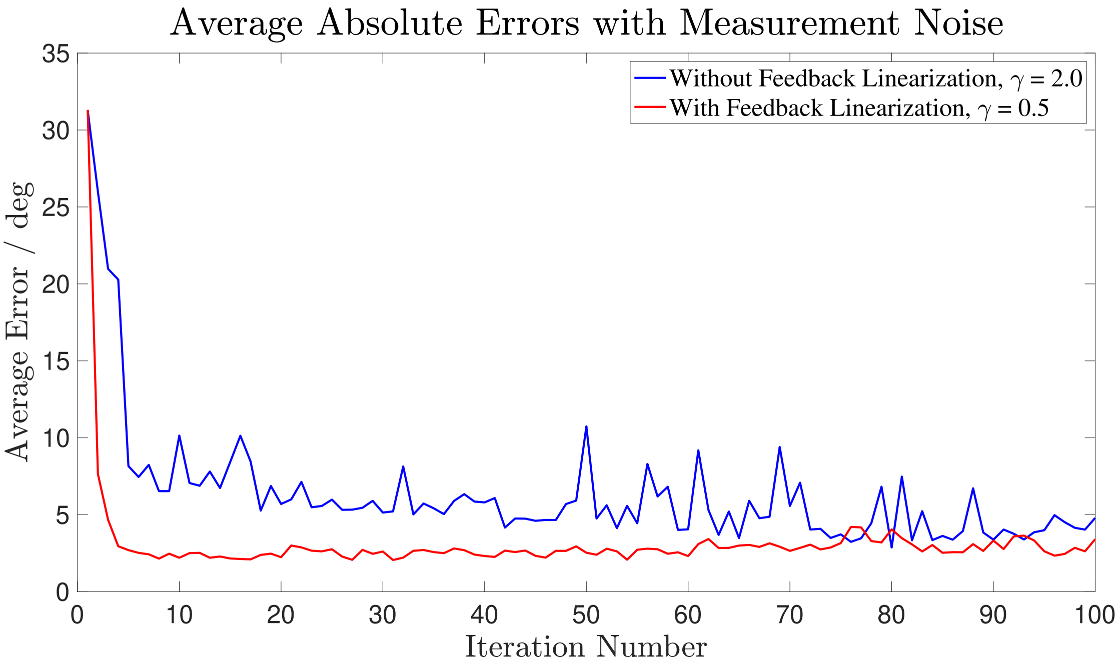

Figure 4.17: Average Absolute Errors with Measurement Noise.

## 4.5 NOILC with Feedback Linearization

The concept of NOILC has been introduced in Section 3.2.2. It is an optimal control based ILC algorithm that aims at a better control performance. The NOILC algorithm has demonstrated great advantages in error convergence rate and monotonicity. However, a major drawback is, the algorithm is designed for linear systems. If there exists a certain level of nonlinearity in the system model, the algorithm can only work on the linearized model of the system, while its performance will face an obvious downgrade. For the exoskeleton system defined by (4.16), applying the control protocol given by (3.36) and (3.37) to the linearized model of the system at its working point, the performance of the system is shown in Figure 4.18 and 4.19.

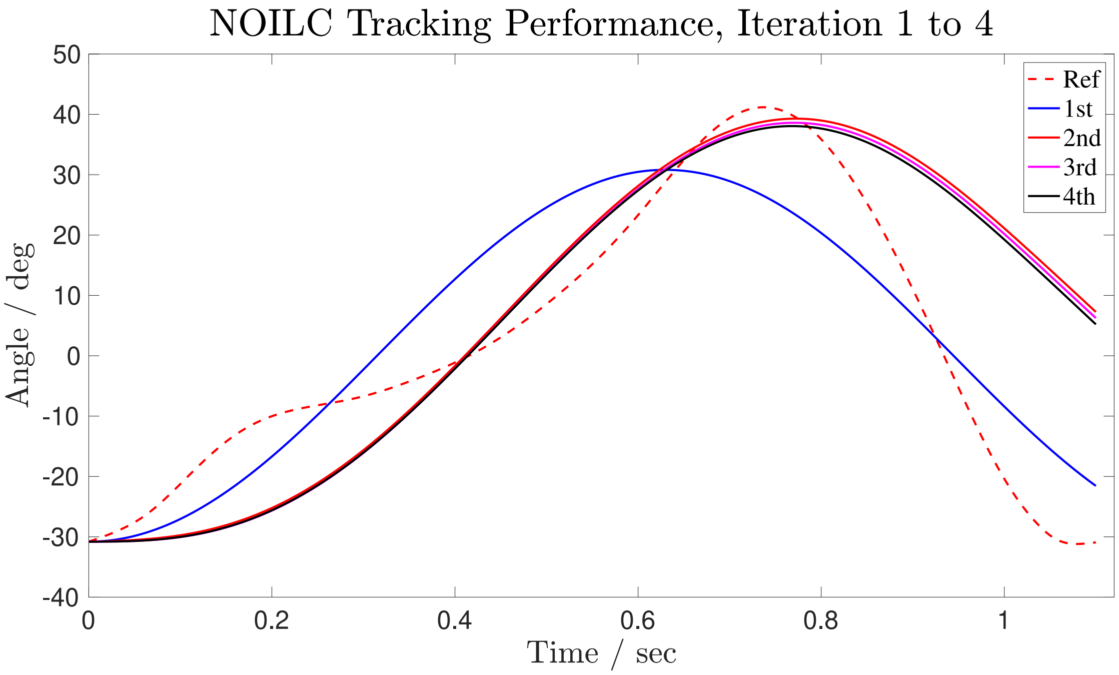

Figure 4.18: Angle Tracking Performance, NOILC, Iteration 1 to 4.

The optimal weights in this case is given by $Q = 70$ and $R = 0.1$. Figure 4.18 is the performance of the system in the first 4 iterations, while Figure 4.19 is the performance

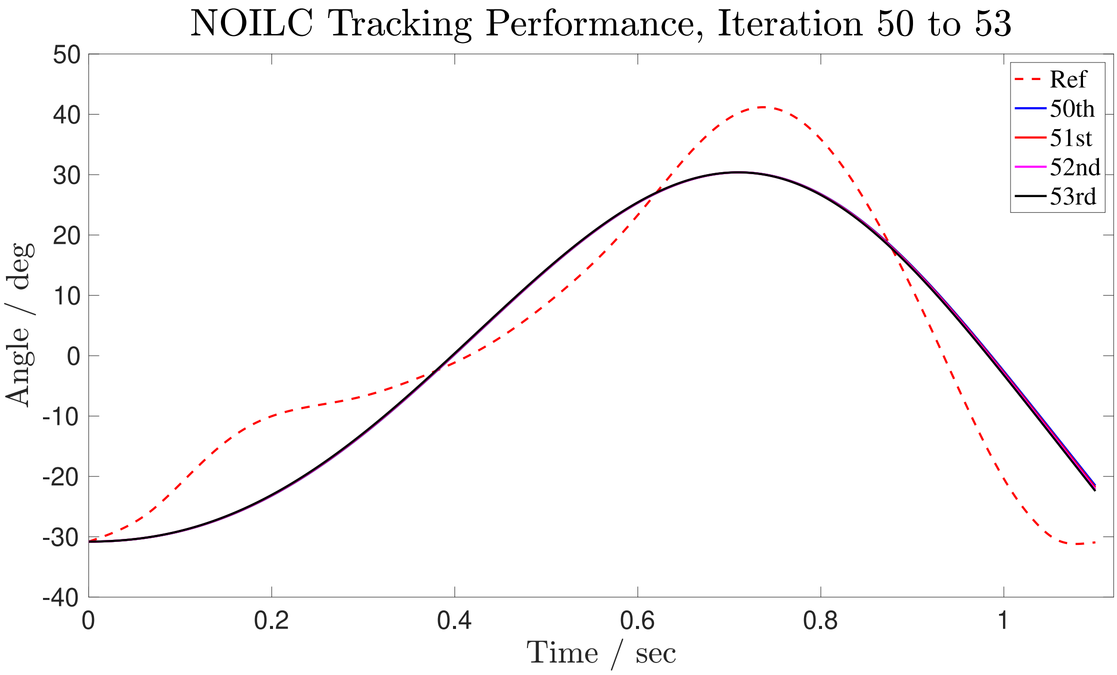

Figure 4.19: Angle Tracking Performance, NOILC, Iteration 50 to 53.

of the system in iteration 50 to 53 when the error has converged to more settled status. As shown in the two figures, due to the effect of nonlinearity, though the system has demonstrated a certain scale of tracking, its performance is far from satisfactory. The error convergence rate is also quite slow. Even after 50 iterations, the error between the reference and the system output is still obvious. The effects of different $Q$ and $R$ values on the average absolute error convergence rate are shown in Figure 4.20 and 4.21. The error convergence rate is relatively slow.

To solve such a problem, the NOILC algorithm can also be improved with a similar feedback linearization way. Design the control law for the feedback linearized system in $k$th iteration as:

$$v_k(t) = -a_1 x_{1,k}(t) - a_2 x_{2,k}(t) + a_1 w_k(t), \tag{4.26}$$

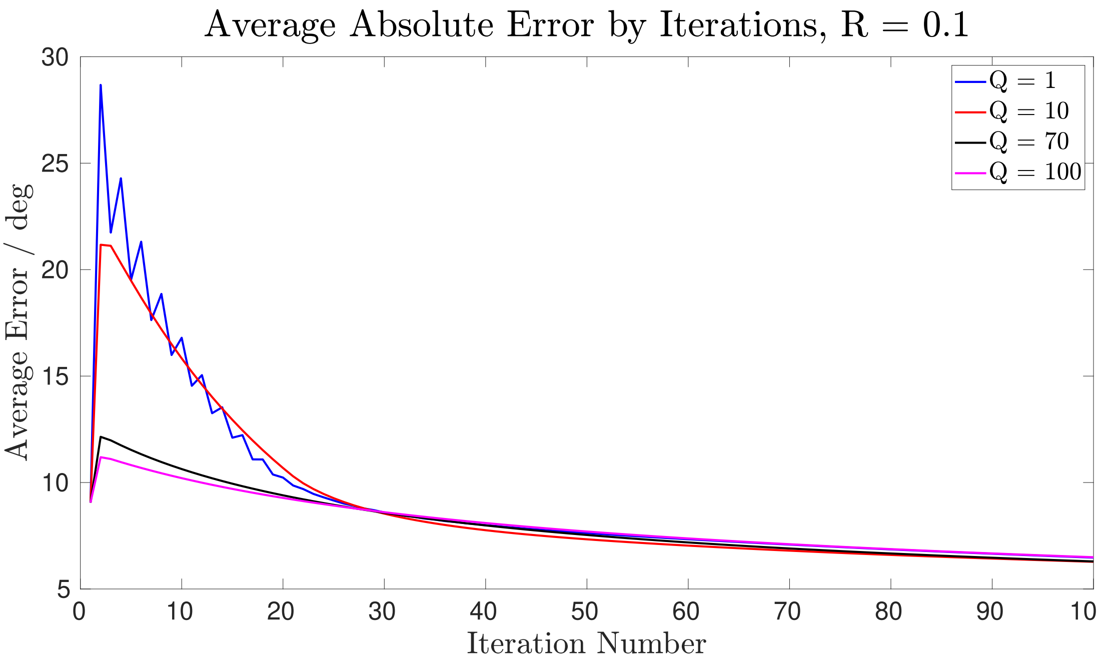

Figure 4.20: Average Absolute Error, NOILC, R = 0.1.

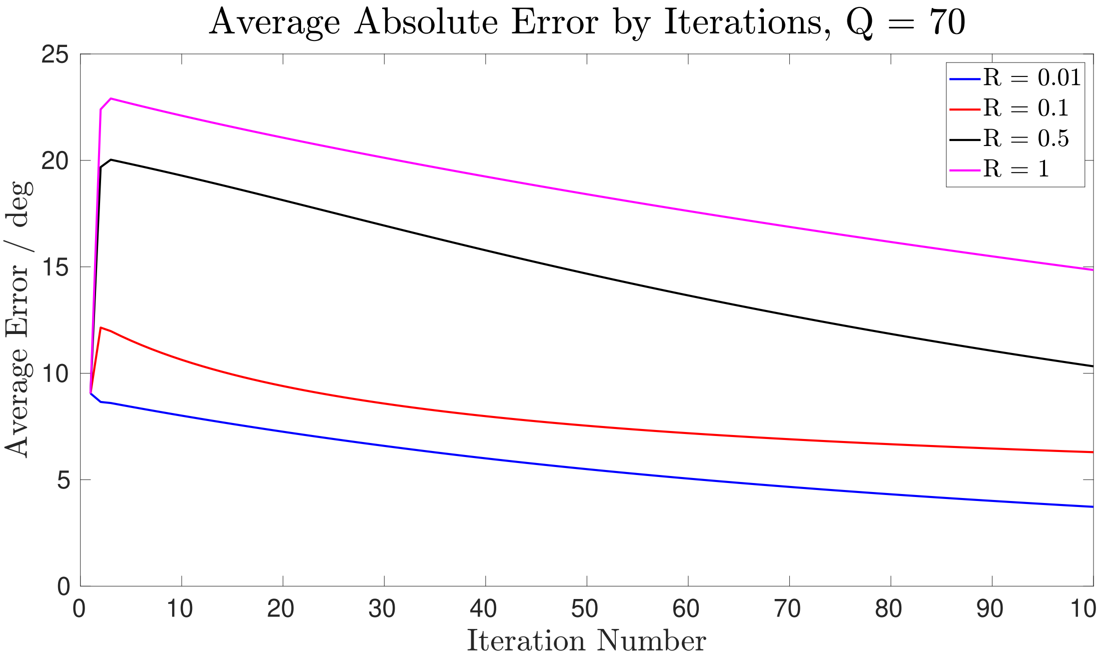

Figure 4.21: Average Absolute Error, NOILC, Q = 70.

where $w_k(t)$ is the NOILC updating law:

$$w_{k+1}(t) = w_k(t) + R^{-1}(t)B^T\phi_{k+1}(t), \tag{4.27}$$

where $\phi_{k+1}(t)$ is the state of the costate system given by (3.37), $R$ is one of the optimal weight, and $B = [0, a_1]$. Then, the final control input to the system (4.16) can also be written in the form of (4.25). But this time, the ILC updating law item $w(t)$ in the control input are changed to the NOILC updating law (4.27).

For the new NOILC algorithm with feedback linearization, its tracking performance is shown in Figure 4.22 and 4.23. Figure 4.22 is the system performance in the first 4 iterations while Figure 4.23 is the performance of the system in iteration 20 to 23 when the error has converged to a more settled status. Compared to the original NOILC algorithm, it can be seen from the figure that for the first few iterations, the system response speed is faster, though the error still exists. The error convergence rate has been optimized on a great scale, as the system almost achieved perfect tracking in only 20 iterations.

The effects of different optimal weight values on the average absolute error convergence rate in the new NOILC algorithm are shown in Figure 4.24 and 4.25. The selection of the optimal weight values has to make a balance between the error convergence rate and system stability. Compared to the results of the original NOILC algorithm, the error convergence of the new proposed NOILC algorithm is now at an exponential rate, which is much faster than the original one. The convergence process is monotonic, which is also very important. The overall performance of the NOILC algorithm has also been significantly improved.

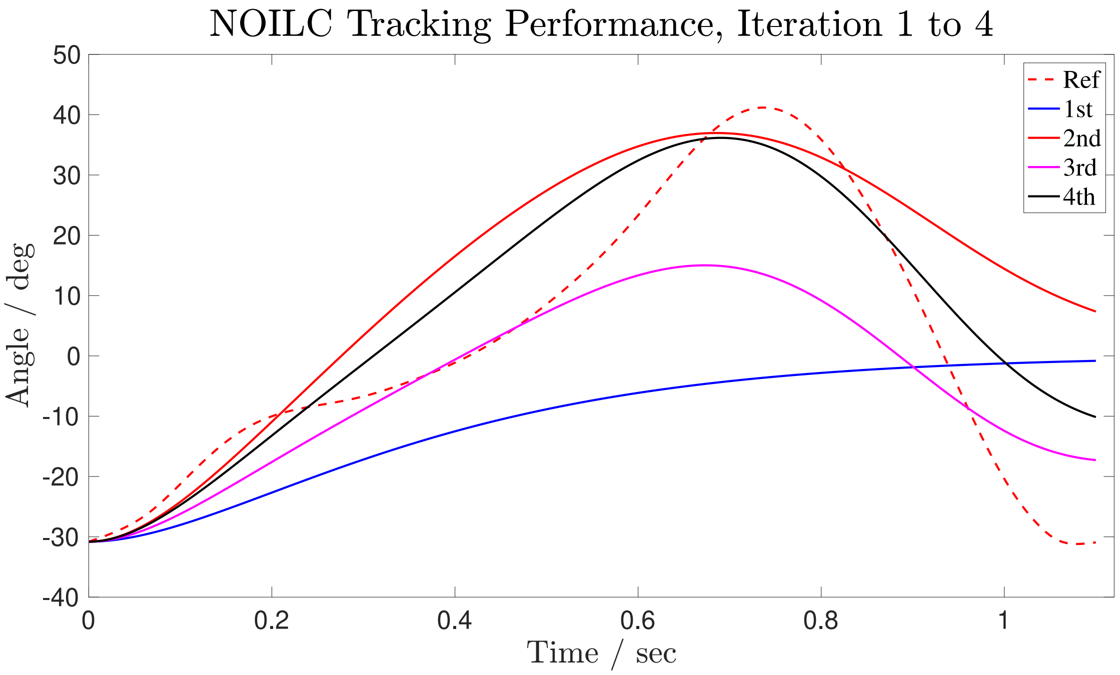

Figure 4.22: Angle Tracking Performance, NOILC with FL, Iteration 1 to 4.

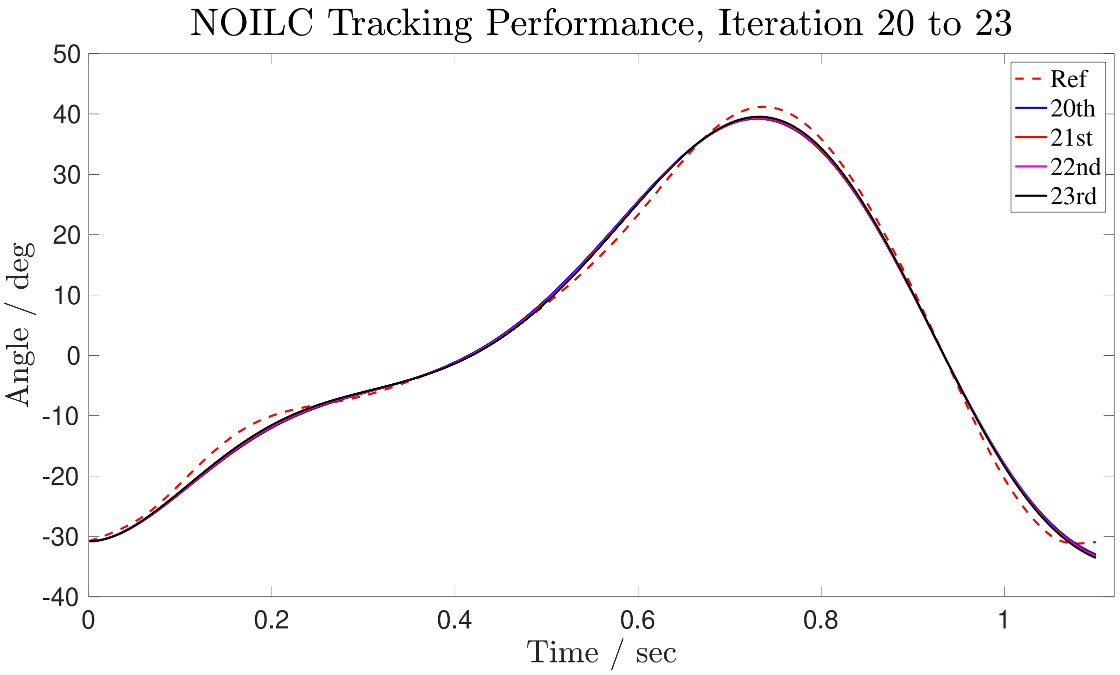

Figure 4.23: Angle Tracking Performance, NOILC with FL, Iteration 20 to 23.

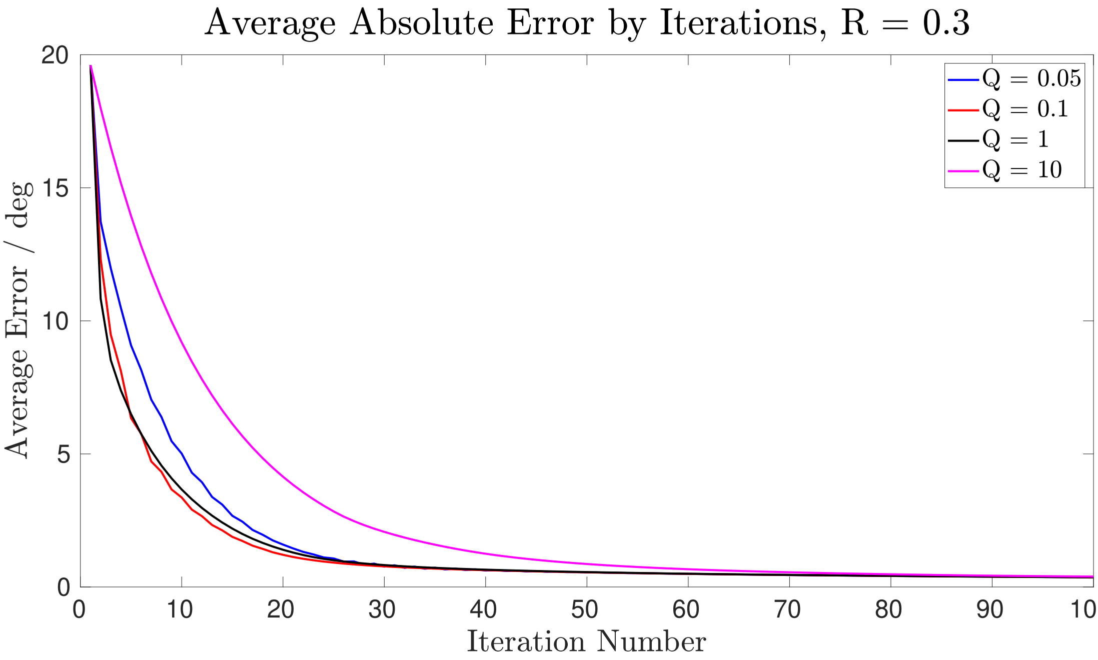

Figure 4.24: Average Absolute Error, NOILC with FL, R = 0.1.

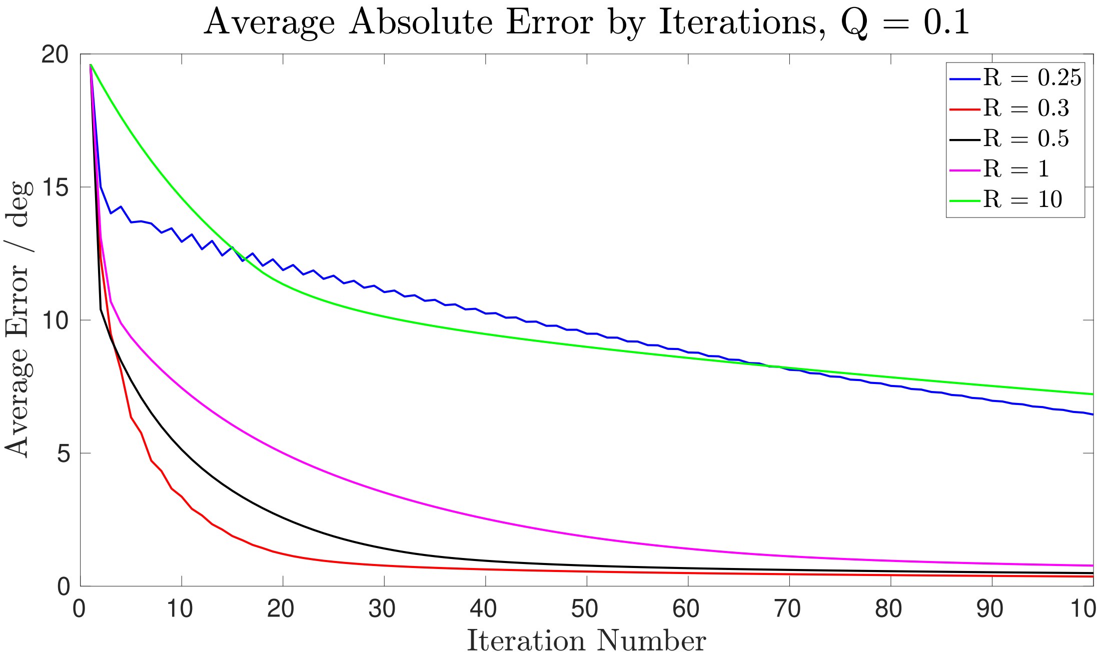

Figure 4.25: Average Absolute Error, NOILC with FL, Q = 70.

## 4.6 Dual Internal Model Structure ILC

Though the system performance has been improved by the new NOILC algorithm, improvement can still be made to obtain a better control result. As can be seen from Figure 4.24, though with an exponential error convergence rate, it still takes the system more than 20 iterations to achieve a small average error. Such a speed may not be enough for some applications. To make further improvements to the system performance, especially in the first few iterations, a DIMS ILC algorithm is introduced.

The DIMS ILC algorithm is an internal-model-based ILC algorithm. In this algorithm, the discrete-time periodical external signal to the system of length $N$ is modeled as an external state-space model given by:

$$\begin{aligned} x_w(i+1) &= A_w x_w(i) + B_w u_w(i), \\ w(i) &= C_w x_w(i) + D_w u_w(i), \end{aligned} \tag{4.28}$$

where $A_w$ is an $N \times N$ matrix, $B_w$ is an $N \times 1$ matrix, and $C_w$ is a $1 \times N$ matrix. For past-error feedforward system, $D_w = 0$. Both periodic reference signal and reference disturbance are included in such an external model. For the example given in later this section, the system has only one output signal and one reference signal, while no other periodical disturbance had been applied to the system. In this case, the values of matrices $A_w$, $B_w$, and $C_w$ is given by:

$$\begin{aligned} A_w &= \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \end{bmatrix}, \\ B_w &= \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}^T, \\ C_w &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}. \end{aligned} \tag{4.29}$$

Then, for a MIMO system, the augmented matrix of the external model is given by:

$$\begin{aligned} A_l &= \mathrm{diag}\{A_w, A_w, \cdots, A_w\},\ B_l = \mathrm{diag}\{B_w, B_w, \cdots, B_w\}, \\ C_l &= \mathrm{diag}\{C_w, C_w, \cdots, C_w\},\ D_l = \mathrm{diag}\{D_w, D_w, \cdots, D_w\}. \end{aligned} \tag{4.30}$$

For the augmented system which contains both the objective system and the external system, its state-space form in $k$th iteration is given by [140]:

$$\begin{bmatrix} x_{l,k}(i+1) \\ x_k(i+1) \end{bmatrix} = \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} \begin{bmatrix} x_{l,k}(i) \\ x_k(i) \end{bmatrix} + \begin{bmatrix} B_l \\ BD_l \end{bmatrix} \tilde{u}_k(i), \tag{4.31}$$

and the output error of the system is given by:

$$e_k(i) = r(i) - \begin{bmatrix} DC_l & C \end{bmatrix} \begin{bmatrix} x_{l,k}(i) \\ x_k(i) \end{bmatrix} - DD_l \tilde{u}_k(i). \tag{4.32}$$

The control input of the state-space system, $\tilde{u}_k(i)$, is given by:

$$\tilde{u}_k(i) = -K_l \begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix}, \tag{4.33}$$

where $\hat{x}_{l,k}(i)$ and $\hat{x}_k(i)$ are state estimates generated by the state observer. The observer is designed as:

$$\begin{aligned} \begin{bmatrix} \hat{x}_{l,k}(i+1) \\ \hat{x}_k(i+1) \end{bmatrix} &= \begin{bmatrix} A_l & 0 \\ BC_l & A \end{bmatrix} \begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix} + \begin{bmatrix} B_l \\ BD_l \end{bmatrix} K_l \begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix} \\ &+ L_l \left( e_k(i) + \left( \begin{bmatrix} DC_l & C \end{bmatrix} - DD_l K_l \right) \begin{bmatrix} \hat{x}_{l,k}(i) \\ \hat{x}_k(i) \end{bmatrix} \right). \end{aligned} \tag{4.34}$$

The $K_l$ and $L_l$ are the control gain and observer gain to be designed. Its value can be designed via the Linear Quadratic Regulator (LQR) method.

Applying such a DIMS ILC algorithm to the shank exoskeleton system, its tracking performance is shown in Figure 4.26 and 4.27. Figure 4.26 is the system performance in the first 4 iterations while Figure 4.27 is the performance of the system in iteration 10 to 13 when the error has converged to a more settled status. Compared to the NOILC algorithms introduced in the previous section, it can be witnessed that the system response speed has been improved on a great scale, while the final error has also been reduced. The system output can follow the reference in less than 5 iterations. After 10 iterations, only a minor error still exists.

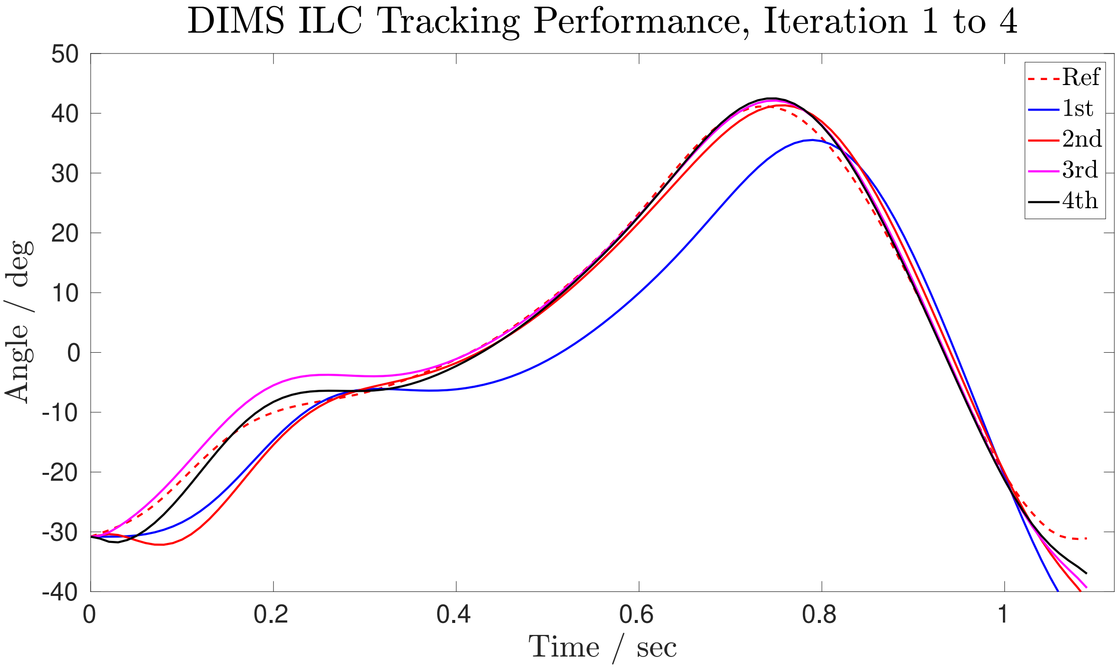

Figure 4.26: Angle Tracking Performance, DIMS ILC, Iteration 1 to 4.

Though the system performance with the DIMS ILC algorithm is already much better than the previous ILC methods, tracking errors can still be witnessed in the first few iterations as well as later iterations. Such a phenomenon is also due to the nonlinearities existing in the system model. To cancel the effect of such nonlinerities and achieve an ideal tracking performance, the proposed feedback linearization ILC method has also been applied to the DIMS ILC method. The performance of the new DIMS ILC algorithm are shown in Figure 4.28 and 4.29. As demonstrated in the figures, the system response is even faster while the output achieves a very precise tracking in less than 3

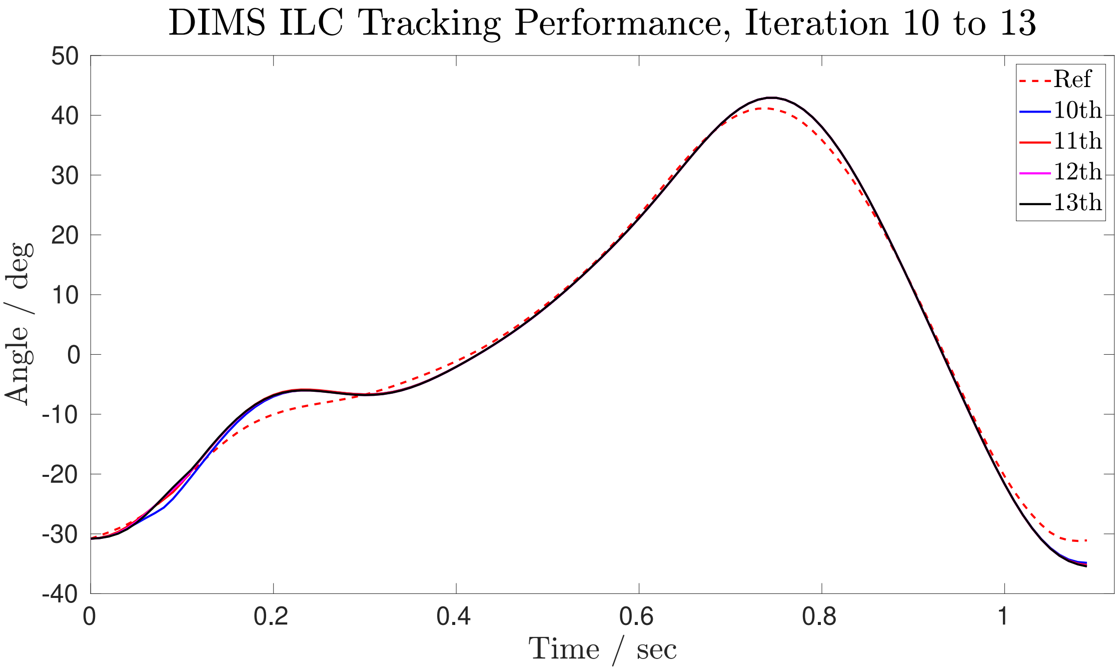

Figure 4.27: Angle Tracking Performance, DIMS ILC, Iteration 10 to 13.

iterations. Meanwhile, the final error has almost been eliminated after 10 iterations.

The effect of the feedback linearization method on DIMS ILC is shown in Figure 4.30. As shown in the figure, the tracking error has been decreased on a great scale. The system response in the first few iterations is very fast and the final tracking error is minor.

The robustness of the DIMS ILC algorithm is not good enough to eliminate the effect of disturbances, but its performance is still better than the original ILC algorithm. For the same Gaussian measurement noise added to the system output, the DIMS ILC algorithm has a faster error convergence rate and a slightly smaller average absolute error. But the system performance is also noisy since this is the kind of random noise that is not good for ILC applications. For a fixed impulse disturbance added to the system output which will appear repeatedly in each iteration, the DIMS ILC algorithm can compensate with

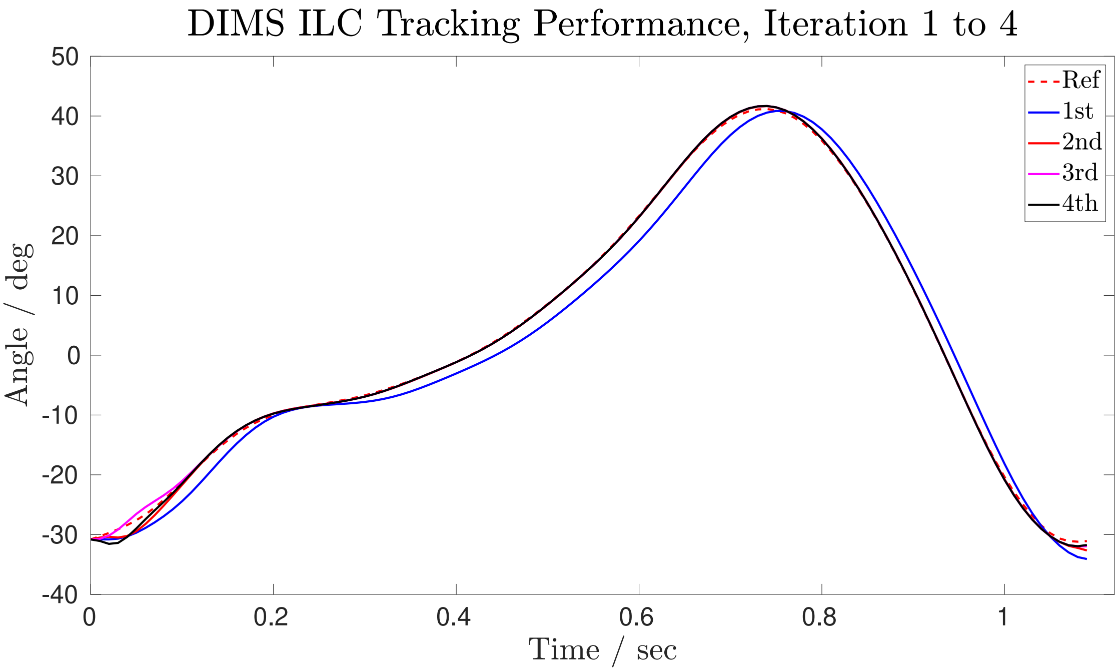

Figure 4.28: Angle Tracking Performance, DIMS ILC with FL, Iteration 1 to 4.

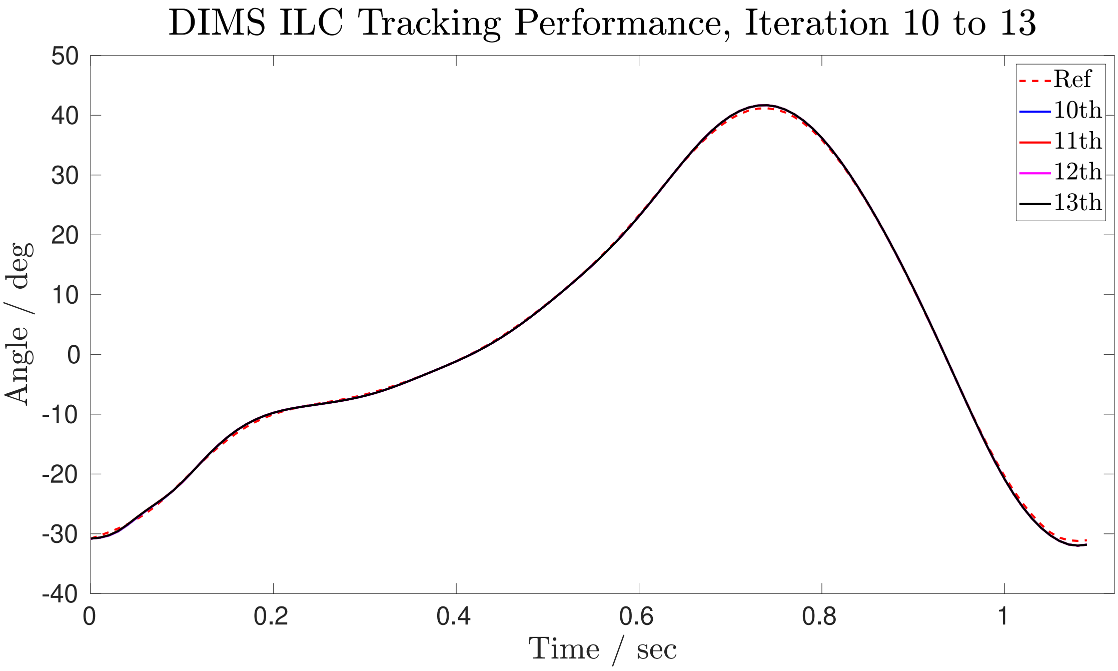

Figure 4.29: Angle Tracking Performance, DIMS ILC with FL, Iteration 10 to 13.

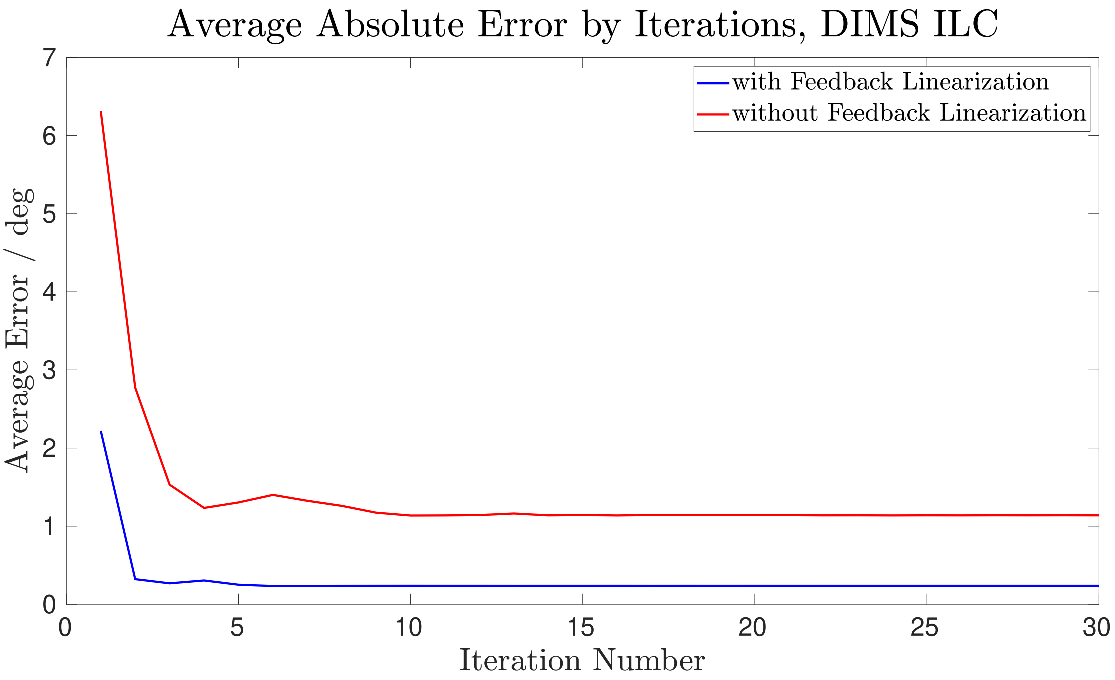

Figure 4.30: Average Absolute Errors, DIMS ILC, with and without FL.

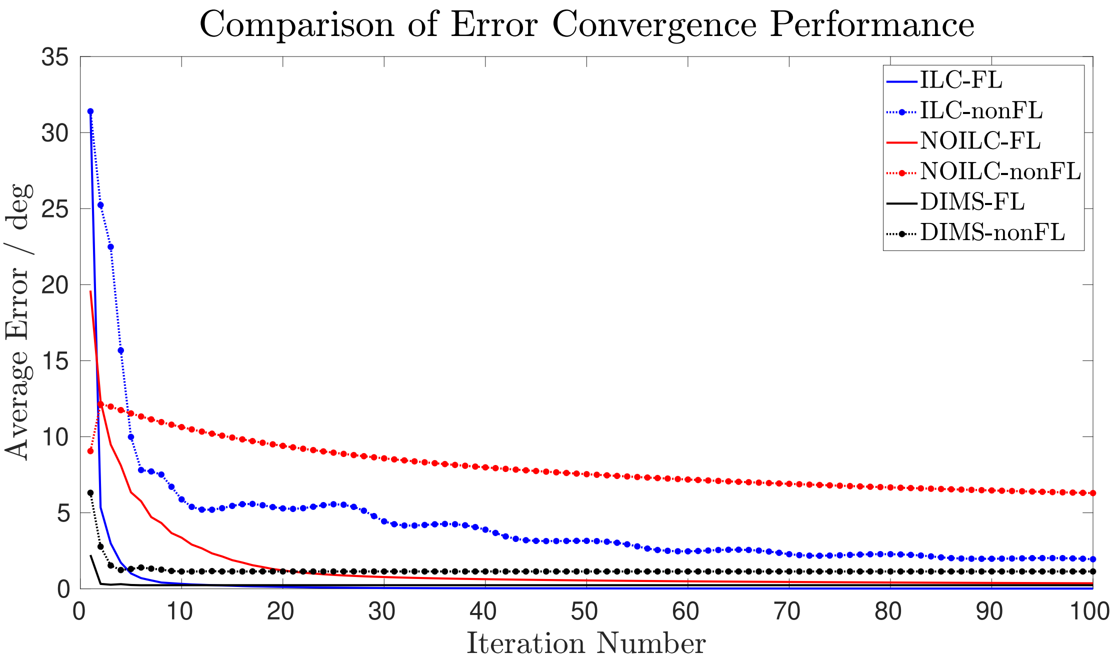

Figure 4.31: Comparison of Error Convergence Rate.

it. The result has a higher average error but can still achieve a good tracking performance. For the system uncertainties that exist during the modelling of the system, the DIMS ILC algorithm faces a slow down in error convergence speed, while its final tracking accuracy has almost not been affected. For the unmodelled dynamics that existed in the system model, the algorithm goes unstable during the test, which leaves the problem worthwhile for further studies.

Finally, the comparison of performance based on error convergence performance for all the ILC algorithms introduced in this chapter is shown in Figure 4.31. As demonstrated in the figure, the DIMS ILC method has the fastest error convergence rate and smallest final error. But the construction of the internal model needs a certain level of system knowledge. The NOILC algorithm is relatively slower in convergence speed, but usually, it can guarantee the monotonic of its error convergence, which is a very important property of ILC algorithms. The D-type ILC has neither the fastest convergence rate nor a guarantee of monotonic. However, it has the simplest control structure and does not need any previous knowledge of the objective system. For all three kinds of ILC algorithms, the proposed feedback linearization ILC method has greatly improved their tracking performance and error convergence rate. The proposed ILC algorithm also enhanced the abilities of these ILC algorithms in dealing with the system nonlinearities.

## 4.7 Conclusion

In this chapter, the control problem for the shank part of the lower limb exoskeleton robot has been discussed. With the periodical gait reference signal and the dynamic model obtained from its equation of motion, the control problem of the shank part lower limb exoskeleton robot has been transformed to a reference tracking problem in a fixed finite time interval. Several ILC algorithms have been applied to solve such a problem.

But due to the unstable and nonlinear features that existed in the dynamic model, their performances are not satisfying. Then, with the proposed novel algorithm, the performances of those ILC algorithms have been greatly improved. Finally, the simulation results of the proposed algorithms indicate that the error convergence rates have been increased for those algorithms.

# Chapter 5

# Adaptive Oscillator Control for Lower Limb Exoskeleton

Other than the ILC methods, the adaptive oscillator method is another method that can be applied to solve the control problem of the exoskeleton robots. This chapter introduces the principle of the adaptive oscillator method and its application on the exoskeleton robot. Section 5.1 presents the structure and principle of the adaptive oscillator. Section 5.2 gives its simulation result on the exoskeleton robot model. A comparison between the ILC methods and the adaptive oscillator method has been represented in Section 5.3. Finally, Section 5.4 summarizes the chapter.

## 5.1 Adaptive Oscillator

In general, an oscillator can be regarded as a dynamic system with a stable limit cycle [108]. By introducing a frequency variable $\omega$, which can determine the intrinsic oscillation frequency of the system, the system can be transferred to an adaptive oscillator. In the meantime, the dynamics of variable $\omega$ are designed to be affected by the external target frequency signal $f$. Then, the oscillation frequency and phase of the designated adaptive oscillator can gradually adapt to the external target frequency signal $f$.

One of the most commonly used adaptive oscillators is the Hopf oscillator. The typical dynamics of the Hopf oscillator is given by [163]:

$$\begin{aligned}\dot{x}_o(t) &= \lambda(\mu - x_o(t)^2 - y_o(t)^2)x_o(t) - \omega(t)y_o(t) + \epsilon f(t),\\ \dot{y}_o(t) &= \lambda(\mu - x_o(t)^2 - y_o(t)^2)y_o(t) + \omega(t)x_o(t),\\ \dot{\omega}(t) &= -\epsilon f(t)\frac{y_o(t)}{\sqrt{x_o(t)^2 + y_o(t)^2}},\end{aligned} \tag{5.1}$$

where $x_o$ and $y_o$ are state variables of the Hopf oscillator, $\mu$ is the oscillation amplitude parameter, $\epsilon$ is the coupling strength parameter, and $\lambda$ determines the recovery speed of the oscillator after perturbations.

The Hopf oscillator has an important property that it can not only imitate the frequency of the external signal but also keep oscillating at such frequency after the external signal disappears. For the Hopf oscillator presented by (5.1), if $f(t)$ is a frequency signal, the oscillation frequency of the Hopf oscillator will gradually synchronize to the frequency of $f(t)$. Then, after the oscillation has started, even if the external signal $f(t)$ returns to zero at some point, the oscillator can still maintain its oscillating frequency. If the input signal is the amplitude error sequence between the external signal and the Hopf oscillator, the Hopf oscillator not only can synchronize with the frequency of the external signal but also can imitate its frequency. Such a property lead to the design of the coupled adaptive oscillator system and the adaptive oscillator control scheme. To better explain the working status of the Hopf oscillator, an example has been given. Setting the parameters in the Hopf oscillator (5.1) as $\mu = 0.2$, $\epsilon = 0.9$, $\lambda = 0.2$, and the initial intrinsic frequency of the oscillator as $\omega_0 = 5\mathrm{rad/s}$, giving a simple external periodical signal:

$$f = \sin(8t), \tag{5.2}$$

which is active to the system between $t = 10$ and $t = 40$, the frequency variation of the adaptive oscillator in this example is shown in Figure 5.1 and 5.2, where Figure 5.1 is in frequency scale and Figure 5.2 is in amplitude scale.

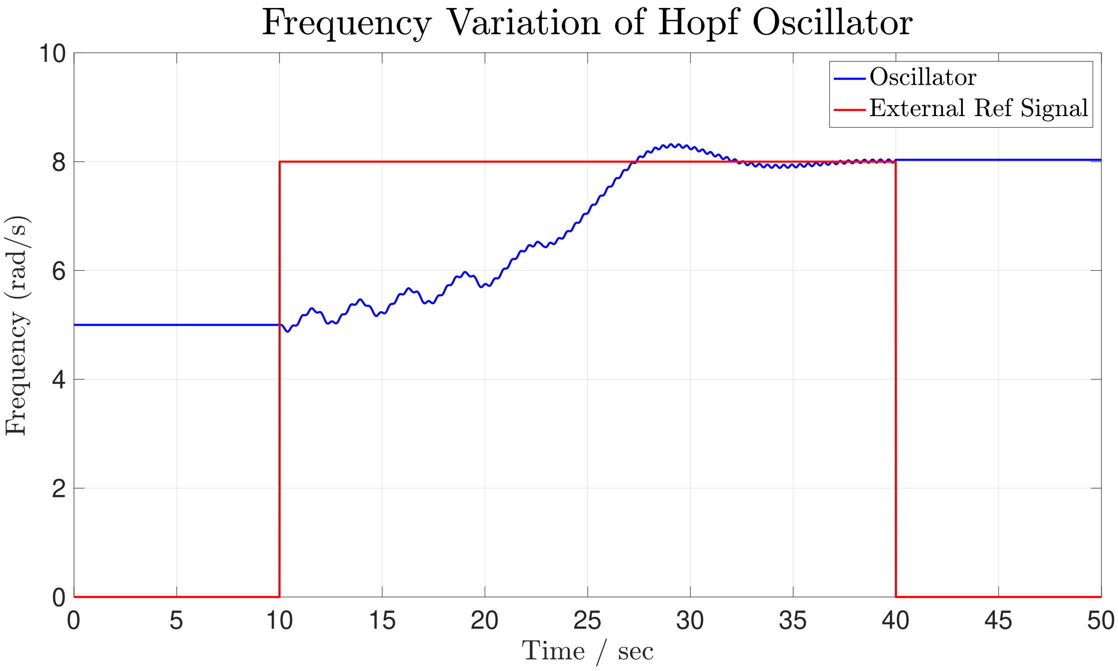

Figure 5.1: Adaptation Process of Hopf Oscillator, Frequency Scale.

As indicated in the figures, the red line represents the external periodical signal while the blue line represents the Hopf oscillator. The initial frequency of the Hopf oscillator is $5\mathrm{rad/s}$. After the external signal $f$ has been activated, the frequency of the Hopf oscillator gradually approaches the frequency of $f$. Then, after the external signal has been turned off at $t = 40$, the Hopf oscillator still keeps its frequency learned from the external signal.

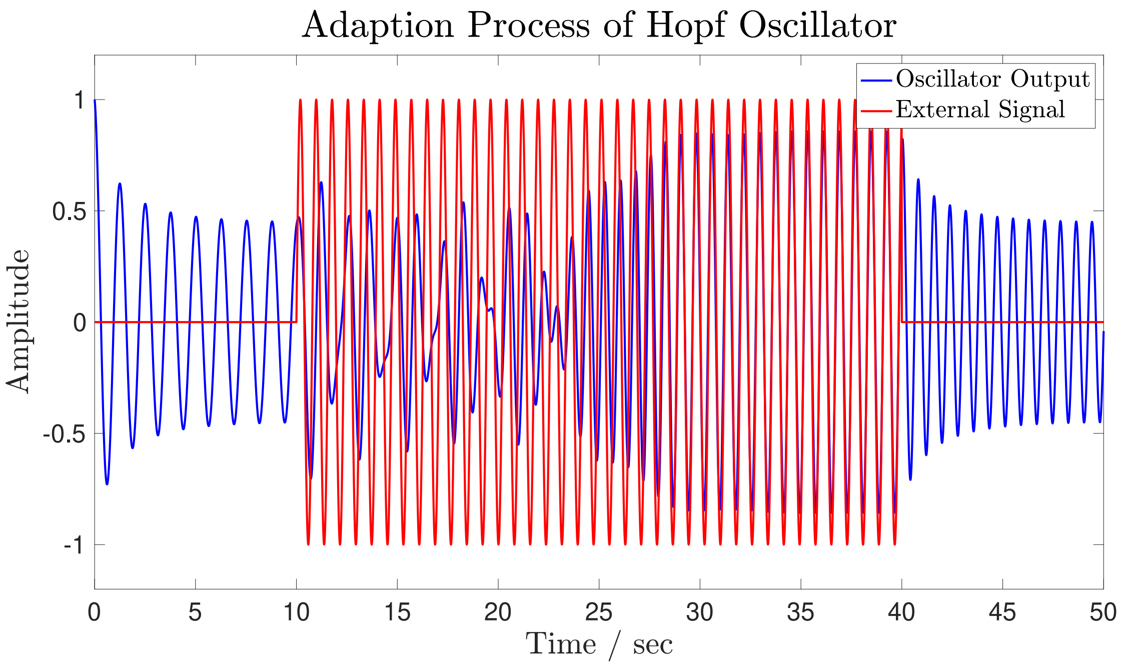

Figure 5.2: Adaptation Process of Hopf Oscillator, Amplitude Scale.

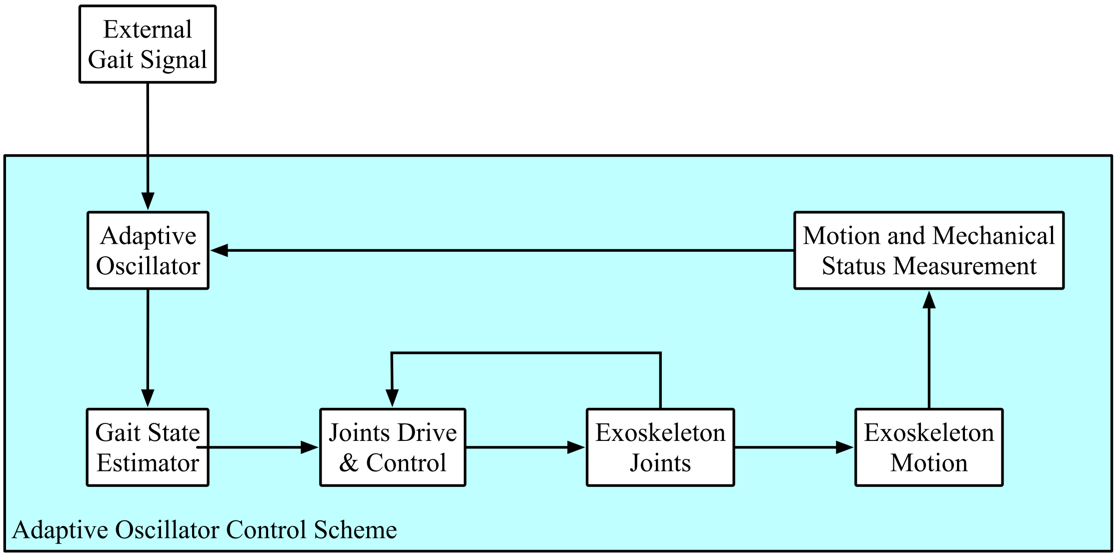

Figure 5.3: Control Scheme of the Adaptive Oscillator Control System.

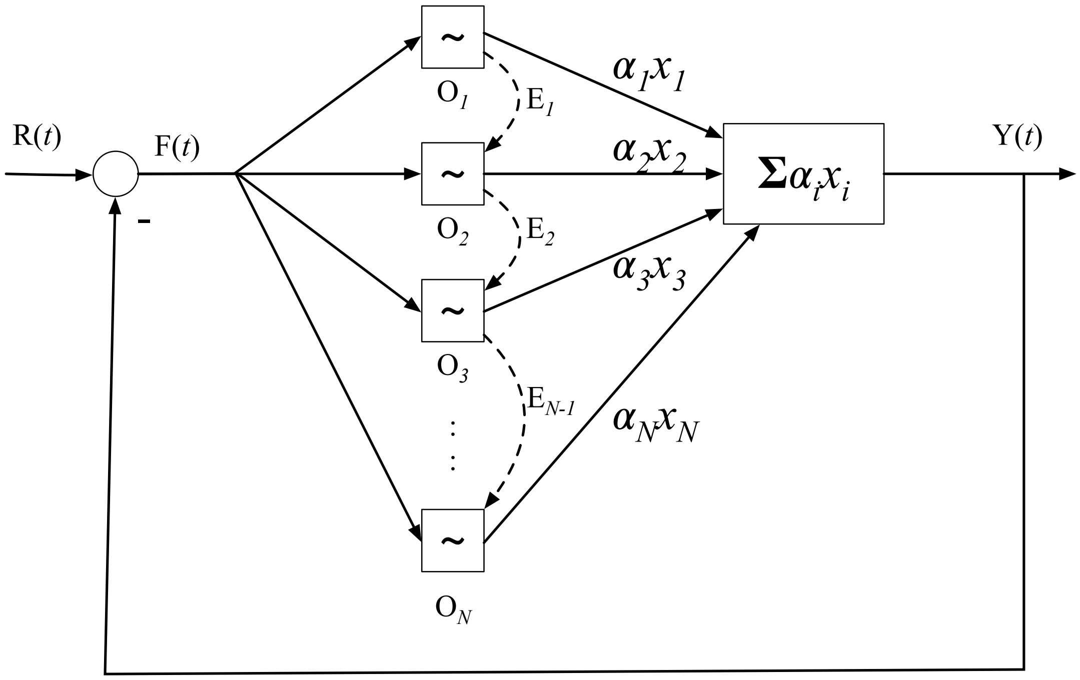

Figure 5.4: Structure of Coupled Adaptive Oscillator System.

Oscillators are said to be coupled if they have interacted with each other. Coupled oscillator system can not only study the frequency of the external signal but also generate designated patterns. The control scheme of the exoskeleton system based on the coupled adaptive oscillator system is shown by Figure 5.3. The structure of the coupled oscillator in the control scheme is shown by Figure 5.4. The adaptive oscillator will learn the frequency and amplitude of the external reference gait signal, then generate estimations of the gait states which can help calculate the assistance torques of the joint actuators. Then new estimations can be generated based on the difference between the exoskeleton motion and the external gait signal. Also, due to the property mentioned in previous paragraphs, the adaptive oscillator can still keep its oscillating frequency and drive the motion of the exoskeleton even if the external reference gait signal has been disconnected.

For the coupled oscillator system in Figure 5.4, $R(t)$ is the external periodic reference signal. $Y(t)$ is the output of the coupled oscillator system, which is a weighted sum of the individual oscillator outputs. $F(t)$ is the error between the external periodic reference signal and the oscillator system output. $O_i$ represents the $i$th individual oscillator in the system where $0 < i < N$. $E_i$ is the remaining error between the reference signal and the output sum of the first $i$ oscillators, where $0 < i < N-1$. A negative feedback loop is also added to the system so that the amplitude of the system output can track the amplitude of the external signal. The dynamics of the $i$th individual oscillator $O_i$ in Figure 5.4 is given by:

$$\begin{aligned}
\dot{x}_{oi}(t) &= \lambda(\mu - x_{oi}(t)^2 - y_{oi}(t)^2)x_{oi}(t) - \omega_i(t)y_{oi}(t) + \epsilon f_i(t) + \eta\sin(\theta_i - \phi_i), \\
\dot{y}_{oi}(t) &= \lambda(\mu - x_{oi}(t)^2 - y_{oi}(t)^2)y_{oi}(t) + \omega_i(t)x_{oi}(t), \\
\dot{\omega_i}(t) &= -\epsilon f_i(t)\frac{y_{oi}(t)}{\sqrt{x_{oi}(t)^2 + y_{oi}(t)^2}},
\end{aligned} \tag{5.3}$$

where $x_{oi}$ and $y_{oi}$ are state variables of the $i$th individual oscillator, $\epsilon$ and $\eta$ are the coupling strength parameters. $f_i$ is the frequency input signal for each individual oscillator, which equals to the sum of $F(t)$ and $E_{N-1}$. $\mu$ is the oscillation amplitude parameter. $\lambda$ determines the recovery speed of the individual oscillator after perturbations. $\omega_i$ is the oscillation frequency of $i$th individual oscillator. $\theta_i$ is the current phase of the $i$th individual oscillator given by:

$$\theta_i = \operatorname{sgn}(x_{oi}) \cos^{-1}(-\frac{y_{oi}}{\sqrt{x_{oi}(t)^2 + y_{oi}(t)^2}}), \tag{5.4}$$

while $\phi_i$ is the phase difference between the $i$th oscillator and 1st oscillator whose dynamics is given by:

$$\dot{\phi}_i = \sin(\frac{\omega_i}{\omega_1}\theta_1 - \theta_i - \phi_i). \tag{5.5}$$

Finally, the dynamics of the weight factor $\alpha_i$ in Figure 5.4 is given by:

$$\dot{\alpha}_i = \kappa x_{oi} f_i(t), \tag{5.6}$$

where $\kappa$ is the learning gain. Due to the property of the adaptive oscillator which can maintain oscillating frequency after the disconnection of external signal, the system can finally reach steady status when $F(t)$ and $f_i(t)$ all equals to zero. In this case, the output of the coupled oscillator system will equal to the external reference signal $R(t)$, which means the system gait has been fully estimated.

## 5.2 Simulation Result

In the exoskeleton robot case, the coupled adaptive oscillators can adapt to the external reference signal with multiple frequency harmonics. For the two continuous joint angle references shown in Figure 3.9 and 3.10, the tracking performances of the coupled oscillator system are shown in Figure 5.5 and 5.6. The two cases represent these two separate sets of gait data that have been tested.

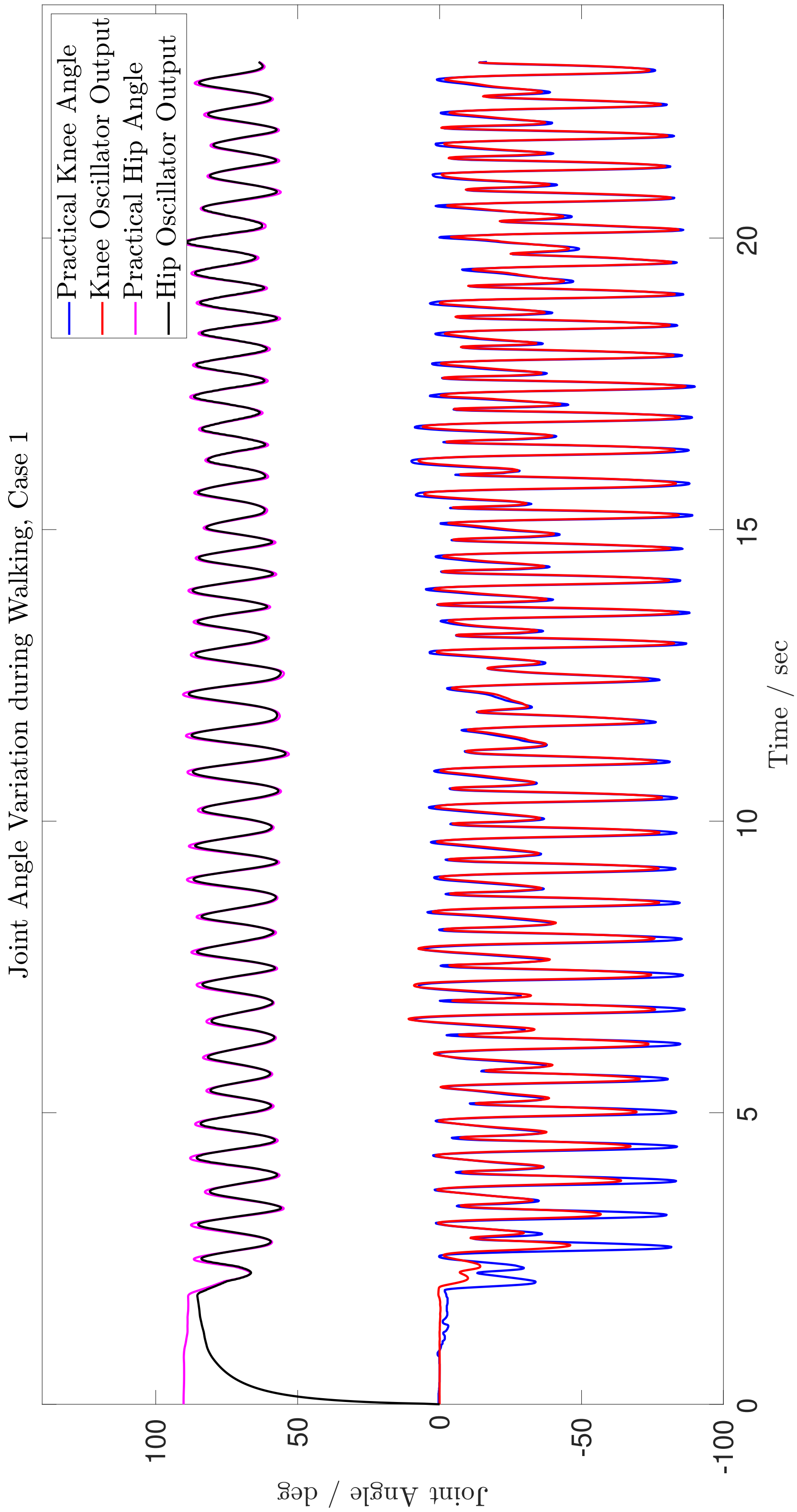

Figure 5.5: Joint Angle Tracking Performance, AO Case 1.

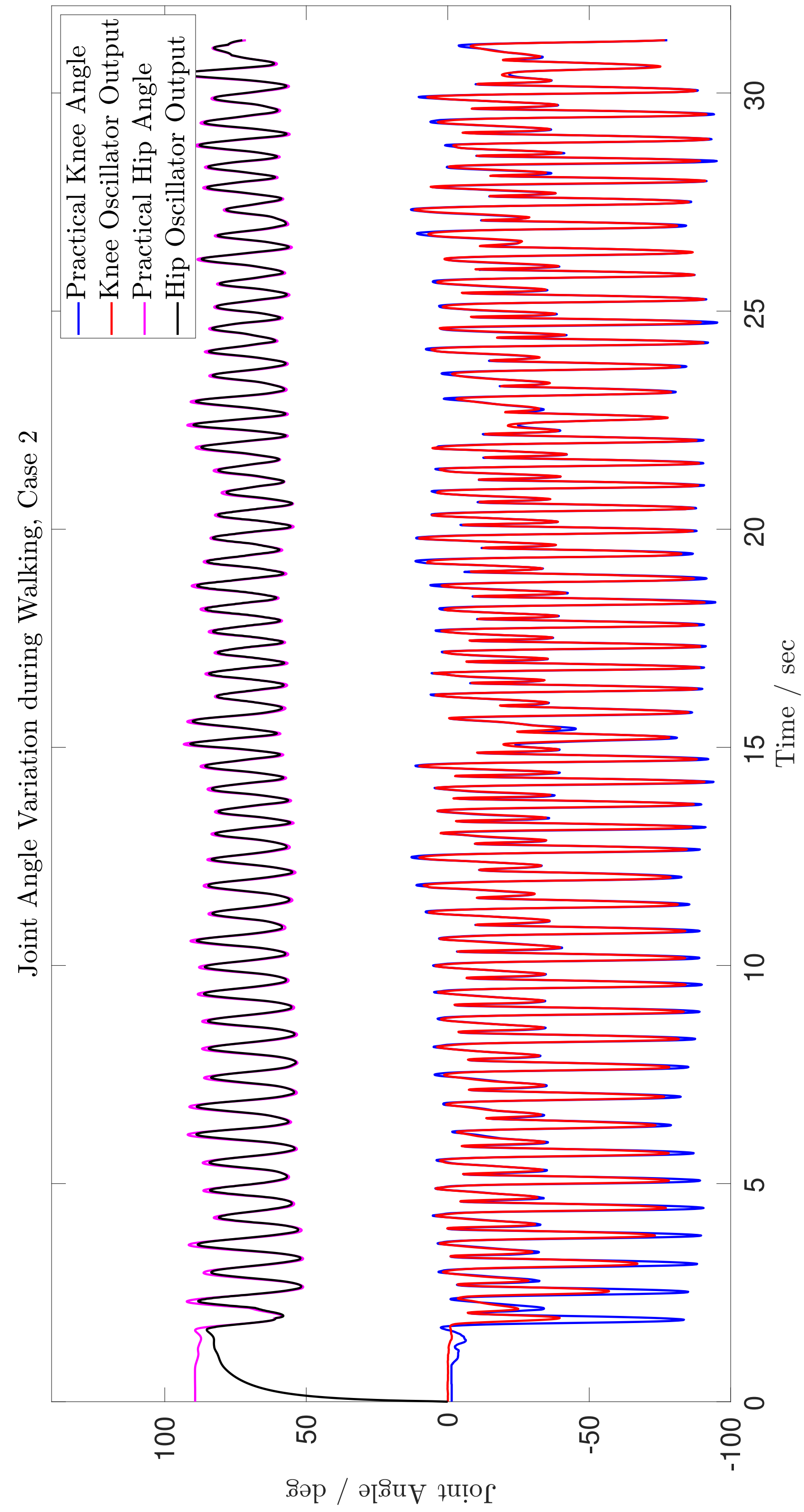

Figure 5.6: Joint Angle Tracking Performance, AO Case 2.

As shown in the figures, the coupled oscillator system has demonstrated an impressive performance of tracking even if the human gait data is not strictly regulated and periodical. The errors are obvious in the first few steps but soon reduced to a much smaller value.

## 5.3 Comparison Between ILC and Adaptive Oscillator Methods

The convergence of the average absolute errors by steps for the adaptive oscillator method is shown in Figure 5.7. A comparison between the performance of the ILC methods and the adaptive oscillator method has also been made.

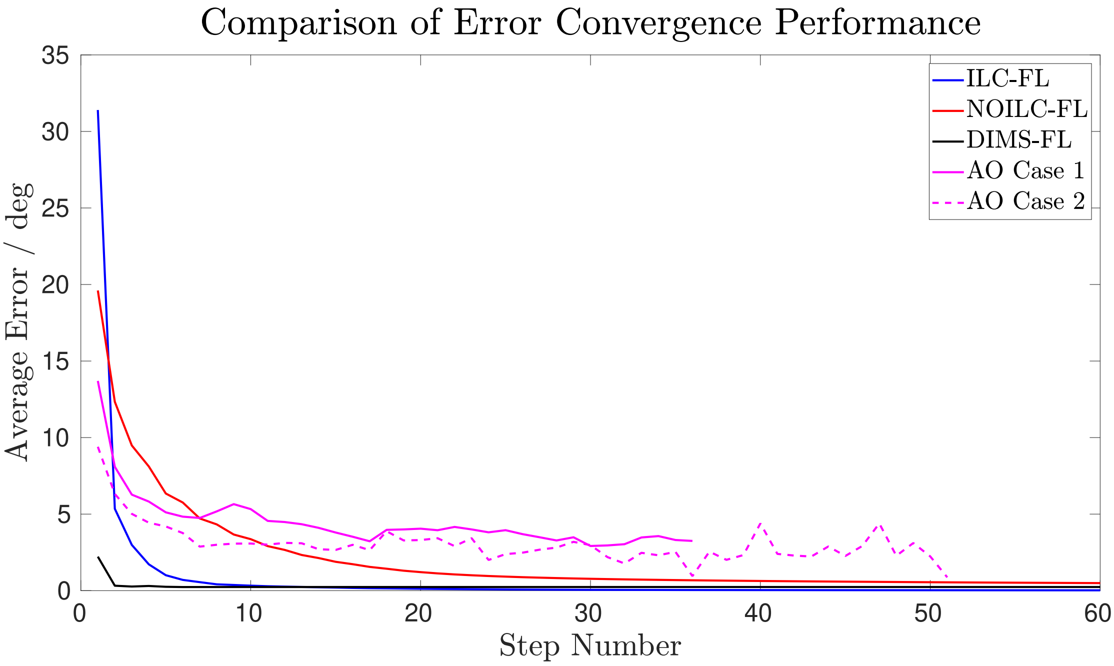

Figure 5.7: Comparison of Error Convergence Performance.

As demonstrated in the figure, case 1 of the adaptive oscillator method has 36 steps in total while case 2 has 51 steps. The angle tracking performance converges at a relatively fast speed in the first few steps, then fluctuates around a small value. Compared to the results of the ILC methods, their error convergence rates are at an equivalent level, but the final error of the adaptive oscillator method is larger than the ILC methods. Meanwhile, since the adaptive oscillator method is tested on an unregulated reference, it is clear that the adaptive oscillator has a better performance in dealing with non-standard gait patterns. The performances of the ILC algorithms will face an obvious downgrade in such cases.

## 5.4 Conclusion

This chapter gives a brief explanation of the principle and structure of the adaptive oscillators. The online learning and fast adaption properties make the adaptive oscillator method suitable for the motion control problem of the exoskeleton robot like the ILC methods. After giving relative simulation results, a comparison between the adaptive oscillator method and the ILC methods has been presented as well. The results show that the adaptive oscillator method has advantages in dealing with non-standard gait patterns but disadvantages in tracking accuracy and error convergence rate. These differences lead to their divergence in application scenarios.

# Chapter 6

# Conclusions and Future Works

This chapter concludes the entire thesis, then indicated the possible directions for future works.

## 6.1 Conclusions

In this thesis, the motion control problem for the lower limb exoskeleton robot has been studied. Firstly, by studying the cyclic property of the human gait, the motion control problem of the exoskeleton robot is converted to a tracking problem on a series of fixed and finite time intervals where every interval represents a gait cycle. Then, a dynamic model of the lower limb exoskeleton robot is established. A novel iterative learning control algorithm has been proposed to deal with the nonlinearities in the model and solve the tracking problem.

Compared to normal iterative learning algorithms, the proposed algorithm has considerable advantages in response speed, tracking accuracy, and error convergence rate. It can also better handle the nonlinearities that exist in the dynamic model of the system. By applying the proposed algorithm to other iterative learning algorithms, including the norm optimal iterative learning control algorithm and the dual internal model structure iterative learning control algorithm, improvements in system performances have been witnessed in both cases. For the exoskeleton model given in the thesis, the response speed has been improved by more than 40 percent, the average absolute error in the steady-state has been reduced by more than 60 percent, and the error convergence speed has been improved from more than 20 iterations to less than 5 iterations. All those examples have demonstrated the effectiveness of the proposed algorithm.

Finally, another method feasible for the motion control problem, the adaptive oscillator method, has also been introduced. The simulation results of the adaptive oscillator method and the iterative learning control methods have been compared as well. The iterative learning control methods also have more advantages in tracking accuracy and error convergence rate, while the adaptive oscillator method is superior in its adaptability to unregulated gait patterns.

## 6.2 Future Research

Along with the remaining works, the following open problems can be heuristic research topics for the iterative learning control of exoskeleton robots in the future.

1. The transition dynamics. During the control process of the exoskeleton robot, it is possible for the wearer to change the gait pattern. The iterative learning controller will be interrupted from its current control process and learn the new gait pattern. The transition dynamics of the controller during such a process can be further studied in the future.

2. More flexible gait cycle length. For the current iterative learning controller, it is assumed that the time length of each gait cycle is the same. Though such an assumption holds in rehabilitation applications, it is possible for the wearer to have a more flexible gait cycle length in other working modes of the exoskeleton. The way to handle such a non-constant gait cycle length can also be a topic for future studies.

3. Robustness. The DIMS ILC method is good at dealing with some periodical disturbances defined by some models. But it is hard to deal with non-periodical disturbances or some disturbances which cannot be modelled. The NOILC method is also weak against non-periodical disturbances. How to enhance the robustness of the iterative learning controller in those aspects is worth future studies.

# Bibliography

[1] M. Gurven and C. Von Rueden, “Hunting, social status and biological fitness,” *Social biology*, vol. 53, no. 1-2, pp. 81–99, 2006.

[2] M. Amann, “Pulmonary system limitations to endurance exercise performance in humans,” *Experimental physiology*, vol. 97, no. 3, pp. 311–318, 2012.

[3] E. M. Abdel-Rahman and M. S. Hefzy, “Three-dimensional dynamic behaviour of the human knee joint under impact loading,” *Medical engineering & physics*, vol. 20, no. 4, pp. 276–290, 1998.

[4] G. Bao, L. Pan, H. Fang, X. Wu, H. Yu, S. Cai, B. Yu, and Y. Wan, “Academic review and perspectives on robotic exoskeletons,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 11, pp. 2294–2304, 2019.

[5] N. G. Hockstein, C. Gourin, R. Faust, and D. J. Terris, “A history of robots: from science fiction to surgical robotics,” *Journal of robotic surgery*, vol. 1, no. 2, pp. 113–118, 2007.

[6] K. Anam and A. A. Al-Jumaily, “Active exoskeleton control systems: State of the art,” *Procedia Engineering*, vol. 41, pp. 988–994, 2012.

[7] C. Yang, J. Zhang, Y. Chen, Y. Dong, and Y. Zhang, "A review of exoskeleton-type systems and their key technologies," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 222, no. 8, pp. 1599–1612, 2008.

[8] R. Bogue, "Exoskeletons and robotic prosthetics: a review of recent developments," *Industrial Robot: an international journal*, 2009.

[9] J. Rosen, M. Brand, M. B. Fuchs, and M. Arcan, "A myosignal-based powered exoskeleton system," *IEEE Transactions on systems, Man, and Cybernetics-part A: Systems and humans*, vol. 31, no. 3, pp. 210–222, 2001.

[10] H. Kawamoto and Y. Sankai, "Power assist system hal-3 for gait disorder person," in *International Conference on Computers for Handicapped Persons*. Springer, 2002, pp. 196–203.

[11] S. K. Banala, S. H. Kim, S. K. Agrawal, and J. P. Scholz, "Robot assisted gait training with active leg exoskeleton (alex)," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 17, no. 1, pp. 2–8, 2008.

[12] A. B. Zoss, H. Kazerooni, and A. Chu, "Biomechanical design of the berkeley lower extremity exoskeleton (bleex)," *IEEE/ASME Transactions on mechatronics*, vol. 11, no. 2, pp. 128–138, 2006.

[13] B. Kalita, J. Narayan, and S. K. Dwivedy, "Development of active lower limb robotic-based orthosis and exoskeleton devices: A systematic review," *International Journal of Social Robotics*, pp. 1–19, 2020.

[14] V. Kumar, Y. V. Hote, and S. Jain, "Review of exoskeleton: history, design and control," in *2019 3rd International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE)*. IEEE, 2019, pp. 677–682.

[15] J. A. de la Tejera, R. Bustamante-Bello, R. A. Ramirez-Mendoza, and J. Izquierdo-Reyes, "Systematic review of exoskeletons towards a general categorization model proposal," *Applied Sciences*, vol. 11, no. 1, p. 76, 2021.

[16] P. Maurice, J. Čamernik, D. Gorjan, B. Schirrmeister, J. Bornmann, L. Tagliapietra, C. Latella, D. Pucci, L. Fritzsche, S. Ivaldi *et al.*, "Objective and subjective effects of a passive exoskeleton on overhead work," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 152–164, 2019.

[17] R. P. Matthew, E. J. Mica, W. Meinhold, J. A. Loeza, M. Tomizuka, and R. Bajcsy, "Introduction and initial exploration of an active/passive exoskeleton framework for portable assistance," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2015, pp. 5351–5356.

[18] T. Bosch, J. van Eck, K. Knitel, and M. de Looze, "The effects of a passive exoskeleton on muscle activity, discomfort and endurance time in forward bending work," *Applied ergonomics*, vol. 54, pp. 212–217, 2016.

[19] W. Van Dijk, H. Van der Kooij, and E. Hekman, "A passive exoskeleton with artificial tendons: Design and experimental evaluation," in *2011 IEEE International Conference on Rehabilitation Robotics.* IEEE, 2011, pp. 1–6.

[20] S. Toxiri, A. S. Koopman, M. Lazzaroni, J. Ortiz, V. Power, M. P. de Looze, L. O'Sullivan, and D. G. Caldwell, "Rationale, implementation and evaluation of assistive strategies for an active back-support exoskeleton," *Frontiers in Robotics and AI*, vol. 5, p. 53, 2018.

[21] M. P. De Looze, T. Bosch, F. Krause, K. S. Stadler, and L. W. O'sullivan, "Exoskeletons for industrial application and their potential effects on physical work load," *Ergonomics*, vol. 59, no. 5, pp. 671–681, 2016.

[22] E. Abele, M. Weigold, and S. Rothenbücher, "Modeling and identification of an industrial robot for machining applications," *CIRP annals*, vol. 56, no. 1, pp. 387–390, 2007.

[23] Y. W. Hong, Y. King, W. Yeo, C. Ting, Y. Chuah, J. Lee, and E.-T. Chok, "Lower extremity exoskeleton: review and challenges surrounding the technology and its role in rehabilitation of lower limbs," *Australian Journal of Basic and Applied Sciences*, vol. 7, no. 7, pp. 520–524, 2013.

[24] R. M. Singh, S. Chatterji, and A. Kumar, "Trends and challenges in emg based control scheme of exoskeleton robots-a review," *Int J Sci Eng Res*, vol. 3, no. 9, pp. 933–940, 2012.

[25] A. J. Young and D. P. Ferris, "State of the art and future directions for lower limb robotic exoskeletons," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 2, pp. 171–182, 2016.

[26] N. Mir-Nasiri and H. S. Jo, "Autonomous low limb exoskeleton to suppress the body weight," in *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2017, pp. 47–51.

[27] B. Chen, H. Ma, L.-Y. Qin, F. Gao, K.-M. Chan, S.-W. Law, L. Qin, and W.-H. Liao, "Recent developments and challenges of lower extremity exoskeletons," *Journal of Orthopaedic Translation*, vol. 5, pp. 26–37, 2016.

[28] A. F. Ruiz-Olaya, A. López, A. F. da Rocha *et al.*, "Upper and lower extremity exoskeletons," in *the book Handbook of Biomechatronics*. Elsevier, 2018, pp. 283–317.

[29] H. Kazerooni, J.-L. Racine, L. Huang, and R. Steger, "On the control of the berkeley lower extremity exoskeleton (bleex)," in *Proceedings of the 2005 IEEE*

*international conference on robotics and automation.* IEEE, 2005, pp. 4353–4360.

[30] G. Zeilig, H. Weingarden, M. Zwecker, I. Dudkiewicz, A. Bloch, and A. Esquenazi, “Safety and tolerance of the rewalk™ exoskeleton suit for ambulation by people with complete spinal cord injury: A pilot study,” *The journal of spinal cord medicine*, vol. 35, no. 2, pp. 96–101, 2012.

[31] O. Cruciger, T. A. Schildhauer, R. C. Meindl, M. Tegenthoff, P. Schwenkreis, M. Citak, and M. Aach, “Impact of locomotion training with a neurologic controlled hybrid assistive limb (hal) exoskeleton on neuropathic pain and health related quality of life (hrqol) in chronic sci: a case study,” *Disability and Rehabilitation: Assistive Technology*, vol. 11, no. 6, pp. 529–534, 2016.

[32] V. Bartenbach, M. Gort, and R. Riener, “Concept and design of a modular lower limb exoskeleton,” in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob).* IEEE, 2016, pp. 649–654.

[33] J. Deng, P. Wang, M. Li, W. Guo, F. Zha, and X. Wang, “Structure design of active power-assist lower limb exoskeleton apal robot,” *Advances in Mechanical Engineering*, vol. 9, no. 11, p. 1687814017735791, 2017.

[34] W. M. Dos Santos, S. L. Nogueira, G. C. de Oliveira, G. G. Peña, and A. A. Siqueira, “Design and evaluation of a modular lower limb exoskeleton for rehabilitation,” in *2017 International Conference on Rehabilitation Robotics (ICORR).* IEEE, 2017, pp. 447–451.

[35] V. Bartenbach, D. Wyss, D. Seuret, and R. Riener, “A lower limb exoskeleton research platform to investigate human-robot interaction,” in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR).* IEEE, 2015, pp. 600–605.

[36] D. J. Hyun, H. Park, T. Ha, S. Park, and K. Jung, "Biomechanical design of an agile, electricity-powered lower-limb exoskeleton for weight-bearing assistance," *Robotics and Autonomous Systems*, vol. 95, pp. 181–195, 2017.

[37] K. Kong and D. Jeon, "Design and control of an exoskeleton for the elderly and patients," *IEEE/ASME Transactions on mechatronics*, vol. 11, no. 4, pp. 428–432, 2006.

[38] S. Wang, L. Wang, C. Meijneke, E. Van Asseldonk, T. Hoellinger, G. Cheron, Y. Ivanenko, V. La Scaleia, F. Sylos-Labini, M. Molinari *et al.*, "Design and control of the mindwalker exoskeleton," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 23, no. 2, pp. 277–286, 2014.

[39] M. S. Amiri, R. Ramli, and M. F. Ibrahim, "Initialized model reference adaptive control for lower limb exoskeleton," *IEEE Access*, vol. 7, pp. 167 210–167 220, 2019.

[40] X. Zhang, W. Jiang, Z. Li, and S. Song, "A hierarchical lyapunov-based cascade adaptive control scheme for lower-limb exoskeleton," *European Journal of Control*, vol. 50, pp. 198–208, 2019.

[41] S. Chen, Z. Chen, B. Yao, X. Zhu, S. Zhu, Q. Wang, and Y. Song, "Adaptive robust cascade force control of 1-dof hydraulic exoskeleton for human performance augmentation," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 589–600, 2016.

[42] H. Kim, Y. J. Shin, and J. Kim, "Design and locomotion control of a hydraulic lower extremity exoskeleton for mobility augmentation," *Mechatronics*, vol. 46, pp. 32–45, 2017.

[43] H. K. Yap, J. H. Lim, F. Nasrallah, J. C. Goh, and R. C. Yeow, "A soft exoskeleton for hand assistive and rehabilitation application using pneumatic actuators

with variable stiffness," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4967–4972.

[44] N. Tsagarakis, D. Caldwell, and G. Medrano-Cerda, "A 7 dof pneumatic muscle actuator (pma) powered exoskeleton," in *8th IEEE International Workshop on Robot and Human Interaction. RO-MAN'99 (Cat. No. 99TH8483)*. IEEE, 1999, pp. 327–333.

[45] A. Zoss and H. Kazerooni, "Design of an electrically actuated lower extremity exoskeleton," *Advanced Robotics*, vol. 20, no. 9, pp. 967–988, 2006.

[46] Ü. Önen, F. M. Botsalı, M. Kalyoncu, M. Tınkır, N. Yılmaz, and Y. Şahin, "Design and actuator selection of a lower extremity exoskeleton," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 623–632, 2013.

[47] D. W. Lee, S. J. Lee, B. R. Yoon, J. Y. Jho, and K. Rhee, "Preliminary study on analysis of pinching motion actuated by electro-active polymers," *International Journal of Mechanical and Mechatronics Engineering*, vol. 8, no. 5, pp. 919–921, 2014.

[48] Y. Bar-Cohen, "Current and future developments in artificial muscles using electroactive polymers," *Expert review of medical devices*, vol. 2, no. 6, pp. 731–740, 2005.

[49] R. Gopura, D. Bandara, K. Kiguchi, and G. K. Mann, "Developments in hardware systems of active upper-limb exoskeleton robots: A review," *Robotics and Autonomous Systems*, vol. 75, pp. 203–220, 2016.

[50] R. Gopura and K. Kiguchi, "Mechanical designs of active upper-limb exoskeleton robots: State-of-the-art and design difficulties," in *2009 IEEE International Conference on Rehabilitation Robotics*. IEEE, 2009, pp. 178–187.

[51] H. S. Lo and S. Q. Xie, “Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects,” *Medical engineering & physics*, vol. 34, no. 3, pp. 261–268, 2012.

[52] C. Nguiadem, M. Raison, and S. Achiche, “Motion planning of upper-limb exoskeleton robots: A review,” *Applied Sciences*, vol. 10, no. 21, p. 7626, 2020.

[53] D. Shi, W. Zhang, W. Zhang, and X. Ding, “A review on lower limb rehabilitation exoskeleton robots,” *Chinese Journal of Mechanical Engineering*, vol. 32, no. 1, pp. 1–11, 2019.

[54] S. Viteckova, P. Kutilek, and M. Jirina, “Wearable lower limb robotics: A review,” *Biocybernetics and biomedical engineering*, vol. 33, no. 2, pp. 96–105, 2013.

[55] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, “Review of assistive strategies in powered lower-limb orthoses and exoskeletons,” *Robotics and Autonomous Systems*, vol. 64, pp. 120–136, 2015.

[56] S. A. Niyogi and E. H. Adelson, “Analyzing gait with spatiotemporal surfaces,” in *Proceedings of 1994 IEEE Workshop on Motion of Non-rigid and Articulated Objects*. IEEE, 1994, pp. 64–69.

[57] A. D. Kuo and J. M. Donelan, “Dynamic principles of gait and their clinical implications,” *Physical therapy*, vol. 90, no. 2, pp. 157–174, 2010.

[58] J. C. Perry and J. Rosen, “Design of a 7 degree-of-freedom upper-limb powered exoskeleton,” in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.* IEEE, 2006, pp. 805–810.

[59] E. M. Arnold and S. L. Delp, “Fibre operating lengths of human lower limb

muscles during walking," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 366, no. 1570, pp. 1530–1539, 2011.

[60] N. Li, L. Yan, H. Qian, H. Wu, J. Wu, and S. Men, "Review on lower extremity exoskeleton robot," *The Open Automation and Control Systems Journal*, vol. 7, no. 1, 2015.

[61] W. Huo, S. Mohammed, Y. Amirat, and K. Kong, "Active impedance control of a lower limb exoskeleton to assist sit-to-stand movement," in *2016 IEEE International Conference on Robotics and Automation (ICRA).* Ieee, 2016, pp. 3530–3536.

[62] J. Zhang, C. C. Cheah, and S. H. Collins, "Experimental comparison of torque control methods on an ankle exoskeleton during human walking," in *2015 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2015, pp. 5584–5589.

[63] J.-H. Kim, M. Shim, D. H. Ahn, B. J. Son, S.-Y. Kim, D. Y. Kim, Y. S. Baek, and B.-K. Cho, "Design of a knee exoskeleton using foot pressure and knee torque sensors," *International Journal of Advanced Robotic Systems*, vol. 12, no. 8, p. 112, 2015.

[64] G. S. Heo, S.-R. Lee, M. K. Kwak, C. W. Park, G. Kim, and C.-Y. Lee, "Motion control of bicycle-riding exoskeleton robot with interactive force analysis," *International Journal of Precision Engineering and Manufacturing*, vol. 16, no. 7, pp. 1631–1637, 2015.

[65] Y. Zeng, J. Yang, and Y. Yin, "Gaussian process-integrated state space model for continuous joint angle prediction from emg and interactive force in a human-exoskeleton system," *Applied Sciences*, vol. 9, no. 8, p. 1711, 2019.

[66] K. Kiguchi, T. Tanaka, and T. Fukuda, “Neuro-fuzzy control of a robotic exoskeleton with emg signals,” *IEEE Transactions on fuzzy systems*, vol. 12, no. 4, pp. 481–490, 2004.

[67] K. Kiguchi and Y. Hayashi, “An emg-based control for an upper-limb power-assist exoskeleton robot,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1064–1071, 2012.

[68] Z. Li, Z. Huang, W. He, and C.-Y. Su, “Adaptive impedance control for an upper limb robotic exoskeleton using biological signals,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1664–1674, 2016.

[69] A. L. Benabid, T. Costecalde, A. Eliseyev, G. Charvet, A. Verney, S. Karakas, M. Foerster, A. Lambert, B. Morinière, N. Abroug *et al.*, “An exoskeleton controlled by an epidural wireless brain–machine interface in a tetraplegic patient: a proof-of-concept demonstration,” *The Lancet Neurology*, vol. 18, no. 12, pp. 1112–1122, 2019.

[70] S. Qiu, Z. Li, W. He, L. Zhang, C. Yang, and C.-Y. Su, “Brain–machine interface and visual compressive sensing-based teleoperation control of an exoskeleton robot,” *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 1, pp. 58–69, 2016.

[71] E. López-Larraz, F. Trincado-Alonso, V. Rajasekaran, S. Pérez-Nombela, A. J. Del-Ama, J. Aranda, J. Minguez, A. Gil-Agudo, and L. Montesano, “Control of an ambulatory exoskeleton with a brain–machine interface for spinal cord injury gait rehabilitation,” *Frontiers in neuroscience*, vol. 10, p. 359, 2016.

[72] M. Witkowski, M. Cortese, M. Cempini, J. Mellinger, N. Vitiello, and S. R. Soekadar, “Enhancing brain-machine interface (bmi) control of a hand exoskeleton using electrooculography (eog),” *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, pp. 1–6, 2014.

[73] W. Meng, Q. Liu, Z. Zhou, Q. Ai, B. Sheng, and S. S. Xie, "Recent development of mechanisms and control strategies for robot-assisted lower limb rehabilitation," *Mechatronics*, vol. 31, pp. 132–145, 2015.

[74] R. Lu, Z. Li, C.-Y. Su, and A. Xue, "Development and learning control of a human limb with a rehabilitation exoskeleton," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3776–3785, 2013.

[75] K. H. Ha, S. A. Murray, and M. Goldfarb, "An approach for the cooperative control of fes with a powered exoskeleton during level walking for persons with paraplegia," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 4, pp. 455–466, 2015.

[76] F. J. Diedrich and W. H. Warren Jr, "The dynamics of gait transitions: Effects of grade and load," *Journal of motor behavior*, vol. 30, no. 1, pp. 60–78, 1998.

[77] Y. Liu, R. Collins, and Y. Tsin, "Gait sequence analysis using frieze patterns," in *European Conference on Computer Vision*. Springer, 2002, pp. 657–671.

[78] Y. Ran, R. Chellappa, and Q. Zheng, "Finding gait in space and time," in *18th international conference on pattern recognition (ICPR'06)*, vol. 4. IEEE, 2006, pp. 586–589.

[79] J. Perry, J. R. Davids *et al.*, "Gait analysis: normal and pathological function," *Journal of Pediatric Orthopaedics*, vol. 12, no. 6, p. 815, 1992.

[80] T. Yamasaki, T. Nomura, and S. Sato, "Phase reset and dynamic stability during human gait," *Biosystems*, vol. 71, no. 1-2, pp. 221–232, 2003.

[81] D. J. Villarreal, H. A. Poonawala, and R. D. Gregg, "A robust parameterization of human gait patterns across phase-shifting perturbations," *IEEE Transactions*

*on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 3, pp. 265–278, 2016.

[82] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic systems*, vol. 1, no. 2, pp. 123–140, 1984.

[83] G. Casalino, "A learning procedure for the control of movements of robotic manipulators," *IASTED Sympo. on Robotics and Automation, 1984*, 1984.

[84] S. Arimoto, "A brief history of iterative learning control," in *Iterative Learning Control*. Springer, 1998, pp. 3–7.

[85] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[86] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1099–1121, 2007.

[87] J. D. Ratcliffe, J. J. Hätönen, P. L. Lewin, E. Rogers, T. J. Harte, and D. H. Owens, "P-type iterative learning control for systems that contain resonance," *International Journal of adaptive control and signal processing*, vol. 19, no. 10, pp. 769–796, 2005.

[88] D. Huang, J.-X. Xu, X. Li, C. Xu, and M. Yu, "D-type anticipatory iterative learning control for a class of inhomogeneous heat equations," *Automatica*, vol. 49, no. 8, pp. 2397–2408, 2013.

[89] Y. Chen and K. L. Moore, "Pi-type iterative learning control revisited," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 3. IEEE, 2002, pp. 2138–2143.

[90] ——, “An optimal design of pd-type iterative learning control with monotonic convergence,” in *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*. IEEE, 2002, pp. 55–60.

[91] A. Madady, “Pid type iterative learning control with optimal gains,” *International Journal of Control, Automation, and Systems*, vol. 6, no. 2, pp. 194–203, 2008.

[92] J. D. Ratcliffe, P. L. Lewin, E. Rogers, J. J. Hatonen, and D. H. Owens, “Norm-optimal iterative learning control applied to gantry robots for automation applications,” *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1303–1307, 2006.

[93] D. R. Yang, K. S. Lee, H. J. Ahn, and J. H. Lee, “Experimental application of a quadratic optimal iterative learning control method for control of wafer temperature uniformity in rapid thermal processing,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 1, pp. 36–44, 2003.

[94] I. Chin, S. J. Qin, K. S. Lee, and M. Cho, “A two-stage iterative learning control technique combined with real-time feedback for independent disturbance rejection,” *Automatica*, vol. 40, no. 11, pp. 1913–1922, 2004.

[95] K. L. Moore, “An iterative learning control algorithm for systems with measurement noise,” in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304)*, vol. 1. IEEE, 1999, pp. 270–275.

[96] Z. Cai, C. T. Freeman, P. L. Lewin, and E. Rogers, “Iterative learning control for a non-minimum phase plant based on a reference shift algorithm,” *Control Engineering Practice*, vol. 16, no. 6, pp. 633–643, 2008.

[97] J.-X. Xu, Q. Hu, T. H. Lee, and S. Yamamoto, “Iterative learning control with smith time delay compensator for batch processes,” *Journal of process Control*, vol. 11, no. 3, pp. 321–328, 2001.

[98] A. Tayebi, S. Abdul, M. Zaremba, and Y. Ye, "Robust iterative learning control design: application to a robot manipulator," *IEEE/ASME Transactions on mechatronics*, vol. 13, no. 5, pp. 608–613, 2008.

[99] M. Norrlof and S. Gunnarsson, "Experimental comparison of some classical iterative learning control algorithms," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 636–641, 2002.

[100] C. T. Freeman, P. L. Lewin, E. Rogers, and J. D. Ratcliffe, "Iterative learning control applied to a gantry robot and conveyor system," *Transactions of the Institute of Measurement and Control*, vol. 32, no. 3, pp. 251–264, 2010.

[101] W. Qian, S. K. Panda, and J.-X. Xu, "Torque ripple minimization in pm synchronous motors using iterative learning control," *IEEE transactions on power electronics*, vol. 19, no. 2, pp. 272–279, 2004.

[102] M. Norrlof, "An adaptive iterative learning control algorithm with experiments on an industrial robot," *IEEE Transactions on robotics and automation*, vol. 18, no. 2, pp. 245–251, 2002.

[103] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.

[104] D. Wang and C. C. Cheah, "An iterative learning-control scheme for impedance control of robotic manipulators," *The International Journal of Robotics Research*, vol. 17, no. 10, pp. 1091–1104, 1998.

[105] J. D. McAuley, "Perception of time as phase: Toward an adaptive-oscillator model of rhythmic pattern processing," Ph.D. dissertation, Indiana University, 1995.

[106] J. Buchli, L. Righetti, and A. J. Ijspeert, "Engineering entrainment and adaptation in limit cycle systems," *Biological Cybernetics*, vol. 95, no. 6, pp. 645–664, 2006.

[107] L. Righetti, J. Buchli, and A. J. Ijspeert, "Adaptive frequency oscillators and applications," *The Open Cybernetics & Systemics Journal*, vol. 3, no. 1, 2009.

[108] T. Nachstedt, C. Tetzlaff, and P. Manoonpong, "Fast dynamical coupling enhances frequency adaptation of oscillators for robotic locomotion control," *Frontiers in neurorobotics*, vol. 11, p. 14, 2017.

[109] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic hebbian learning in adaptive frequency oscillators," *Physica D: Nonlinear Phenomena*, vol. 216, no. 2, pp. 269–281, 2006.

[110] D. Eck, "Finding downbeats with a relaxation oscillator," *Psychological research*, vol. 66, no. 1, pp. 18–25, 2002.

[111] L. Righetti and A. J. Ijspeert, "Programmable central pattern generators: an application to biped locomotion control," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 1585–1590.

[112] R. Ronsse, S. M. M. De Rossi, N. Vitiello, T. Lenzi, M. C. Carrozza, and A. J. Ijspeert, "Real-time estimate of velocity and acceleration of quasi-periodic signals using adaptive oscillators," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 783–791, 2013.

[113] J. Liu, X. Chen, L. Yang, J. Gao, and X. Zhang, "Analysis and compensation of reference frequency mismatch in multiple-frequency feedforward active noise and vibration control system," *Journal of Sound and Vibration*, vol. 409, pp. 145–164, 2017.

[114] J. Buchli, F. Iida, and A. J. Ijspeert, "Finding resonance: Adaptive frequency oscillators for dynamic legged locomotion," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3903–3909.

[115] T. Yan, A. Parri, V. R. Garate, M. Cempini, R. Ronsse, and N. Vitiello, "An oscillator-based smooth real-time estimate of gait phase for wearable robotics," *Autonomous Robots*, vol. 41, no. 3, pp. 759–774, 2017.

[116] J. Buchli, L. Righetti, and A. J. Ijspeert, "Frequency analysis with coupled nonlinear oscillators," *Physica D: Nonlinear Phenomena*, vol. 237, no. 13, pp. 1705–1718, 2008.

[117] G. Aguirre-Ollinger, "Exoskeleton control for lower-extremity assistance based on adaptive frequency oscillators: Adaptation of muscle activation and movement frequency," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 229, no. 1, pp. 52–68, 2015.

[118] T. Xue, Z. Wang, T. Zhang, and M. Zhang, "Adaptive oscillator-based robust control for flexible hip assistive exoskeleton," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3318–3323, 2019.

[119] K. Seo, K. Kim, Y. J. Park, J.-K. Cho, J. Lee, B. Choi, B. Lim, Y. Lee, and Y. Shim, "Adaptive oscillator-based control for active lower-limb exoskeleton and its metabolic impact," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6752–6758.

[120] R. Ronsse, N. Vitiello, T. Lenzi, J. Van Den Kieboom, M. C. Carrozza, and A. J. Ijspeert, "Human–robot synchrony: flexible assistance using adaptive oscillators," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 4, pp. 1001–1012, 2010.

[121] R. Ronsse, S. M. M. De Rossi, N. Vitiello, T. Lenzi, B. Koopman, H. Van Der Kooij, M. C. Carrozza, and A. J. Ijspeert, "Real-time estimate of period derivatives using adaptive oscillators: Application to impedance-based walking assistance," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3362–3368.

[122] T. Xue, Z. Wang, T. Zhang, O. Bai, M. Zhang, and B. Han, "A new delayless adaptive oscillator for gait assistance," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3459–3464.

[123] J. L. Pons, R. Ceres, and F. Pfeiffer, "Multifingered dextrous robotics hand design and control: a review," *Robotica*, vol. 17, no. 6, pp. 661–674, 1999.

[124] J. L. Pons, *Wearable robots: biomechatronic exoskeletons*. John Wiley & Sons, 2008.

[125] D. A. Winter, *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[126] D. H. Owens, "Iterative learning control." 2015.

[127] T. Meng and W. He, "Iterative learning control of a robotic arm experiment platform with input constraint," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 664–672, 2017.

[128] D. Zhang, Z. Wang, and T. Masayoshi, "Neural-network-based iterative learning control for multiple tasks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 9, pp. 4178–4190, 2020.

[129] T. Sugie and T. Ono, "An iterative learning control law for dynamical systems," *Automatica*, vol. 27, no. 4, pp. 729–732, 1991.

[130] J.-Y. Jung, W. Heo, H. Yang, and H. Park, “A neural network-based gait phase classification method using sensors equipped on lower limb exoskeleton robots,” *Sensors*, vol. 15, no. 11, pp. 27 738–27 759, 2015.

[131] M. A. Gomes, G. L. M. Silveira, and A. A. Siqueira, “Gait pattern adaptation for an active lower-limb orthosis based on neural networks,” *Advanced Robotics*, vol. 25, no. 15, pp. 1903–1925, 2011.

[132] S. Han, H. Wang, and Y. Tian, “Adaptive computed torque control based on rbf network for a lower limb exoskeleton,” in *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*. IEEE, 2018, pp. 35–40.

[133] J. E. Kurek and M. B. Zaremba, “Iterative learning control synthesis based on 2-d system theory,” *IEEE Transactions on Automatic control*, vol. 38, no. 1, pp. 121–125, 1993.

[134] J.-X. Xu, S. K. Panda, and T. H. Lee, *Real-time iterative learning control: design and applications*. Springer Science & Business Media, 2008.

[135] N. Amann, D. H. Owens, and E. Rogers, “Iterative learning control for discrete-time systems with exponential rate of convergence,” *IEE Proceedings-Control Theory and Applications*, vol. 143, no. 2, pp. 217–224, 1996.

[136] J. Choi and J. Lee, “Adaptive iterative learning control of uncertain robotic systems,” *IEE Proceedings-Control Theory and Applications*, vol. 147, no. 2, pp. 217–223, 2000.

[137] J.-X. Xu and Z. Qu, “Robust iterative learning control for a class of nonlinear systems,” *Automatica*, vol. 34, no. 8, pp. 983–988, 1998.

[138] B. Wu, E. K. Poh, D. Wang, and G. Xu, "Satellite formation keeping via real-time optimal control and iterative learning control," in *2009 IEEE Aerospace conference*. IEEE, 2009, pp. 1–8.

[139] Y. Wang, F. Gao, and F. J. Doyle III, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[140] C. T. Freeman, M. A. Alsubaie, Z. Cai, E. Rogers, and P. L. Lewin, "A common setting for the design of iterative learning and repetitive controllers with experimental verification," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 3, pp. 230–249, 2013.

[141] C. T. Freeman, E. Rogers, J. H. Burridge, A.-M. Hughes, and K. L. Meadmore, *Iterative learning control for electrical stimulation and stroke rehabilitation*. Springer, 2015.

[142] S.-J. Yu, J.-H. Wu, and X.-W. Yan, "A pd-type open-closed-loop iterative learning control and its convergence for discrete systems," in *Proceedings. International Conference on Machine Learning and Cybernetics*, vol. 2. IEEE, 2002, pp. 659–662.

[143] C.-K. Chen and J. Hwang, "Pd-type iterative learning control for the trajectory tracking of a pneumatic xy table with disturbances," *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, vol. 49, no. 2, pp. 520–526, 2006.

[144] D. H. Owens, "Norm optimal iterative learning control," in *Iterative Learning Control*. Springer, 2016, pp. 233–276.

[145] B. D. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

[146] M. Athans and P. L. Falb, *Optimal control: an introduction to the theory and its applications*. Courier Corporation, 2013.

[147] M. Volckaert, J. Swevers, and M. Diehl, "A two step optimization based iterative learning control algorithm," in *Dynamic Systems and Control Conference*, vol. 44175, 2010, pp. 579–581.

[148] E. Rogers, D. H. Owens, H. Werner, C. T. Freeman, P. L. Lewin, S. Kichhoff, C. Schmidt, and G. Lichtenberg, "Norm optimal iterative learning control with application to problems in accelerator based free electron lasers and rehabilitation robotics," *European Journal of Control*, vol. 16, no. 5, pp. 497–524, 2010.

[149] H. A. Foudeh, P. Luk, and J. Whidborne, "Application of norm optimal iterative learning control to quadrotor unmanned aerial vehicle for monitoring overhead power system," *Energies*, vol. 13, no. 12, p. 3223, 2020.

[150] R. Nagamune and J. Choi, "Parameter reduction of nonlinear least-squares estimates via nonconvex optimization," in *2008 American Control Conference*. IEEE, 2008, pp. 1298–1303.

[151] J. T. Betts, "Solving the nonlinear least square problem: Application of a general method," *Journal of Optimization Theory and Applications*, vol. 18, no. 4, pp. 469–483, 1976.

[152] J.-X. Xu and Y. Tan, *Linear and nonlinear iterative learning control*. Springer, 2003, vol. 291.

[153] J.-X. Xu, "A survey on iterative learning control for nonlinear systems," *International Journal of Control*, vol. 84, no. 7, pp. 1275–1294, 2011.

[154] T.-Y. Kuc, J. S. Lee, and K. Nam, "An iterative learning control theory for a class of nonlinear dynamic systems," *Automatica*, vol. 28, no. 6, pp. 1215–1221, 1992.

[155] M. Norrlöf and S. Gunnarsson, "Time and frequency domain convergence properties in iterative learning control," *International Journal of Control*, vol. 75, no. 14, pp. 1114–1126, 2002.

[156] T. Seel, T. Schauer, and J. Raisch, "Monotonic convergence of iterative learning control systems with variable pass length," *International Journal of Control*, vol. 90, no. 3, pp. 393–406, 2017.

[157] M. Yu and C. Li, "Robust adaptive iterative learning control for discrete-time nonlinear systems with time-iteration-varying parameters," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1737–1745, 2017.

[158] T. D. Son, G. Pipeleers, and J. Swevers, "Robust monotonic convergent iterative learning control," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1063–1068, 2015.

[159] K. Takakusaki, "Functional neuroanatomy for posture and gait control," *Journal of movement disorders*, vol. 10, no. 1, p. 1, 2017.

[160] G. Bovi, M. Rabuffetti, P. Mazzoleni, and M. Ferrarin, "A multiple-task gait analysis approach: kinematic, kinetic and emg reference data for healthy young and adult subjects," *Gait & posture*, vol. 33, no. 1, pp. 6–13, 2011.

[161] J. Apkarian, S. Naumann, and B. Cairns, "A three-dimensional kinematic and dynamic model of the lower limb," *Journal of biomechanics*, vol. 22, no. 2, pp. 143–155, 1989.

[162] Z. Ding, *Nonlinear and Adaptive Control Systems.* Institution of Engineering and Technology, 2013.

[163] A. M. Oluwatosin, K. Djouani, and Y. Hamam, "Theory of adaptive oscillators:

Mathematical principles and background," in *2013 Africon*. IEEE, 2013, pp. 1–6.