Title
High Precision Control of Indirect Drive Systems Based on End-effector Sensor Information

Author
Han, Cheng-Huei

Publication Date
2009

Peer reviewed|Thesis/dissertation

High Precision Control of Indirect Drive Systems Based on End-effector Sensor Information

by

Cheng-Huei Han

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Masayoshi Tomizuka, Chair
Professor John Karl Hedrick
Professor Claire Tomlin

Fall 2009

High Precision Control of Indirect Drive Systems Based on End-effector Sensor Information

by Cheng-Huei Han

Abstract

High Precision Control of Indirect Drive Systems Based on End-effector Sensor Information

by

Cheng-Huei Han

Doctor of Philosophy in Engineering-Mechanical Engineering

University of California, Berkeley

Professor Masayoshi Tomizuka, Chair

This dissertation emphasizes the use of end-effector sensor information for the performance enhancement of indirect drive systems for motion control. Indirect drive systems, unlike direct drive systems, use gear transmission mechanisms in the chain of links and actuator for the purpose of speed reduction and torque amplification. As a result, they are widely used in applications that require high torque capacity such as robotic applications. The gear transmission mechanisms, however, introduce compliance and nonlinear properties such as friction and hysteresis to the system. Moreover, when robots are driven at high speeds to increase productivity and quality, oscillations on the end-effector often occur caused by the transmission mechanism. Thus, the use of gear mechanisms brings great challenges to the design of servo control systems for robot manipulators that requires high precision at high speed.

To enhance the performance of servo control systems for robot manipulators, this dissertation first presents a tuning method that automatically finds the servo gains of fixed structure controllers for a specified trajectory. The current practice is to tune the controller gains manually, which is a time-consuming task even for experienced control engineers. An automated gain tuning process saves not only time but also the labor cost. The tuning method presented in this dissertation finds the optimal controller gains using real-time nonlinear programming. The controller gains are tuned and the effectiveness of the tuning method is demonstrated by experiments. The dissertation then presents an adaptive disturbance cancellation scheme to reduce the oscillations caused by the transmission error from speed reducers. To

enhance performance of the adaptive scheme while maintaining a good transient response, two modifications are introduced to the basic compensation structure. Experimental results confirm the effectiveness of the proposed schemes and the improvement in load side performance.

Robots often perform the same task repeatedly in industrial applications and thus the tracking error becomes repetitive from one run to another. Iterative learning control is a practical and promising method that reduces the error which repeats in every cycle. An optimization-based iterative learning controller design for the purpose of disturbance rejection is proposed in the third part of this dissertation. It is a model-based design method, where the trade-off between performance and robustness can be handled. Two iterative learning controllers based on different sensor information are designed and compared. Due to the lack of load side position measurements, a load side position estimation algorithm based on Kalman filtering is proposed. The experimental results are presented to confirm the effectiveness of the estimation scheme and the benefits of applying learning controllers for rejecting load side vibrations.

Experiments of the research issues mentioned above are performed on a single axis test stand. As the first step to generalize the control algorithms developed in this dissertation to actual multi-degree-of-freedom robots, system identification of the FANUC M-16iB robot is presented in the last part of this dissertation.

_____

Professor Masayoshi Tomizuka
Dissertation Committee Chair

To my mom, and my dad in heaven

Contents

1 Introduction **1**
1.1 Background 1
1.2 Motivation and Contribution 3
1.2.1 Sensor-based Controller Tuning of Indirect Drive Trains . . . 3
1.2.2 Suppression of Oscillations for Indirect drive Robots using Joint Sensing Information 4
1.2.3 Iterative Learning Control for Indirect Drive Trains 5
1.2.4 Identification of Dynamics of a Robot Manipulator 6
1.3 Outline 7

2 Indirect Drive Train System **8**
2.1 Introduction 8
2.2 Single-Joint Indirect Drive Train Experimental Setup 8
2.3 Single-Joint Indirect Drive Model 10
2.4 Model Characteristics 12
2.4.1 Summary 15

3 Sensor-based Controller Tuning of Robot Manipulators by Real-time Optimization **16**
3.1 Introduction 16
3.2 Single-Joint Indirect Drive Train System 18
3.2.1 Model 18
3.2.2 Controller Structure 18
3.3 Controller Tuning Methodology 20
3.3.1 Cost Functions 20
3.3.2 Update Law 21
3.3.3 Estimation of ∇J in Real-Time 21
3.3.4 Controller Tuning Scheme 25
3.4 Controller Tuning Results 25
3.4.1 Iterative Feedback Tuning (IFT) 25
3.4.2 Feedback Controller Tuning 26

3.4.3 Feedforward Controller Tuning 29
3.4.4 Selection of Parameters for the RTNLP method 34
3.5 State Feedback Controller Tuning 37
3.5.1 Observer and State feedback Controller Structure 37
3.5.2 Experimental Results 38
3.6 Summary 40

4 Suppression of Oscillations for Indirect drive Robots using Joint Sensing Information 42
4.1 Introduction 42
4.2 Dynamic Model of Single-Joint Indirect Drive Train with Transmission Error 45
4.3 Transmission Error Compensation Algorithms 47
4.3.1 Adaptive Disturbance Cancellation Scheme 47
4.3.2 Modified Schemes for Performance Enhancement 52
4.4 Experimental Study 55
4.4.1 Experimental Setup 55
4.4.2 Experimental Results 56
4.5 Summary 58

5 Iterative Learning Control for Indirect Drive Trains 61
5.1 Introduction 61
5.2 Matrix Description of the System 62
5.3 An Optimization-based Approach to ILC 64
5.4 Motor Side Iterative Learning Controller 65
5.5 Load Side Iterative Learning Controller 68
5.6 Load Side Position Estimation 70
5.7 Experimental Study 72
5.7.1 Experimental Setup 72
5.7.2 Experimental Results 74
5.8 Summary 79

6 Identification of Dynamics of a Robot Manipulator 80
6.1 Introduction 80
6.2 Robot Instrumentation 81
6.2.1 Hardware Setup 81
6.2.2 Software Setup 83
6.3 SISO System Identification 83
6.3.1 System Modeling 83
6.3.2 System Identification Approach 85
6.3.3 Experimental Procedure 91

6.3.4 Data Processing 92
6.3.5 Results 93
6.4 MIMO System Identification 95
6.4.1 Problem Formulation 95
6.4.2 Identification Approach 96
6.4.3 Experimental Procedure 99
6.4.4 Results 99
6.5 Summary 101

7 Conclusion **104**
7.1 Contributions 104
7.2 Future Research 106

List of Figures

2.1 Single-joint indirect drive train experimental setup 9
2.2 Accelerometer installation [68] 10
2.3 Single-joint indirect drive train 11
2.4 Magnitude of the frequency response of the single-joint indirect drive setup 12
2.5 Static friction identification result 14

3.1 Block diagram of the control system 18
3.2 Block diagram of the feedback controller 19
3.3 Block diagram for gradient estimation 22
3.4 Frequency response of a peak filter 23
3.5 The feedforward controller tuning scheme 25
3.6 Minimization of the cost function in feedback controller tuning experiment 27
3.7 Tuning of PID controller parameters using RTNLP in experiment . 27
3.8 Feedback controller tuning responses before and after 30
3.9 Fourth order trajectory profiles 31
3.10 Load side reference trajectory 32
3.11 Minimization of the cost function in experiment (Feedforward controller tuning) 33
3.12 Tuning of feedforward controller parameters using RTNLP in experiment 33
3.13 Feedforward controller tuning responses before and after 35
3.14 Evolution of the cost function during tuning of feedforward controller parameters using RTNLP with various values of γ ($\gamma_n = diag([5 \times 10^{-15},\ 5 \times 10^{-8},\ 5 \times 10^{-7},\ 5 \times 10^{-8},\ 5 \times 10^{-15},\ 5 \times 10^{-15}])$) 36
3.15 The state feedback controller tuning scheme 37
3.16 Minimization of the cost function in state feedback controller tuning experiment 39
3.17 State feedback controller tuning responses before and after 41

4.1 Harmonic drive gear components [26] 43
4.2 Principles of motion [26] 43
4.3 Definition of the transmission error 44
4.4 Block diagram of a single-joint indirect drive train with transmission error 46
4.5 Block diagram of the closed-loop control system 47
4.6 Load side angular acceleration response before compensation 48
4.7 The proposed adaptive compensation scheme 50
4.8 Block diagram of the proposed speed control system 50
4.9 Frequency response from the disturbance input, d, to the velocity error before and after applying $C_f(s)$ 52
4.10 Comparison of responses from the measurement output and the Kalman filter output 54
4.11 The proposed compensation scheme with the consideration of measurement bias 55
4.12 Load side reference trajectory 56
4.13 Load side acceleration error responses before and after 57
4.14 Load side acceleration error responses considering measurement bias before and after 58
4.15 Estimation of the Fourier series coefficients $\hat{a}(t)$ and $\hat{b}(t)$ 59
4.16 Amplitude spectrum of the load side position error 60

5.1 Controller structure of an indirect drive system with an iterative learning controller 63
5.2 Disturbance observer structure - Motor side ILC 66
5.3 Disturbance observer structure - Load side ILC 69
5.4 Desired load side trajectory 73
5.5 Comparison of the load side position estimation error 75
5.6 Bias estimation of the acceleration measurement 75
5.7 Motor side iterative learning controller results 76
5.8 Load side iterative learning controller results 78
5.9 Comparison of motor side and load side iterative learning controllers 79

6.1 M-16iB FANUC robot 81
6.2 Hardware setup 82
6.3 MATLAB GUI 84
6.4 Robot postures for SISO system identification 87
6.5 System identification flow chart 88
6.6 Controller structure in closed-loop identification 90
6.7 Measured data and curve fitting results for J_1 93
6.8 Closed-loop and open-loop comparison 94

6.9 Controller structure of M-16iB robot system 97
6.10 Example of the time responses for the identification of $G_{11}(s)$ 97
6.11 MIMO system identification procedure for $G_{11}(s)$ 98
6.12 Frequency responses of MIMO system identification from u_1 100
6.13 Frequency responses of MIMO system identification from u_2 102
6.14 Coherence spectrum of MIMO system identification 103
6.15 Curve fitting result for $G_{11}(s)$ 103

List of Tables

2.1 Identified parameters for the single-joint robot 13
2.2 The identified friction parameters (SI Units) 14

3.1 Feedback controller parameters after tuning 28
3.2 Feedforward controller parameters after tuning 34
3.3 Feedback controller parameters for different adaptation speed γ after tuning ($\gamma_n = diag([5 \times 10^{-15},\ 5 \times 10^{-8},\ 5 \times 10^{-7},\ 5 \times 10^{-8},\ 5 \times 10^{-15},\ 5 \times 10^{-15}])$) 37

5.1 The noise variance used in experiments 72

6.1 Joint angles for SISO system identification 86
6.2 Feedback parameters for SISO system identification 91
6.3 Torque amplitudes for SISO system identification 92
6.4 Identified system parameters 95

Acknowledgement

Looking back the past four years in Berkeley, it is so true for me to say, there is no way for me to make it to this point without so many people's help. About two and half years ago, I was suffering from personal and family issues and struggling with the pressure from school work and research. When I finally decided to give up on everything I have worked so far and ran away from the stressful reality, a wise man encouraged and taught me a big lesson in my life. He is my advisor, Professor Masayoshi Tomizuka. He taught me how to face the difficulties and have faith in myself. I would like to express my deepest gratitude and respect to him for his continuous encouragement, guidance, and good advices as a mentor, teacher, and as a person. If it were not him and Mrs. Tomizuka, I would not be writing the acknowledgement today.

I would also like to thank Prof. J. Karl Hedrick and Prof. Claire Tomlin for their kindness, and support through the important moments of my years in Berkeley. Moreover, I am greatly thankful for their effort and patience to help me finish my dissertation. FANUC Ltd. has been the sponsor for the work in this dissertation. I would like to thank Dr. Seiuemon Inaba, and Dr. Yoshiharu Inaba sincerely for their continuing support and encouragement and Mr. Seigou Katou for his technical help that made my dissertation possible.

I want to thank all the previous and current members in the Mechanical Systems Control laboratory. I want to give special thanks to my robotics group members, Haifei Cheng, Wenjie Chen, Pedro Reynoso, Mike Chan, Dr. Kiyonori Inaba, Dr. Soo Joen, and Dr Chun-Chih Wang for their professional opinions, creative ideas, and the emotional encouragements. I also want to thank Emma Yu, Hoday Sterns, Nancy Dong, Lucy Fan, Chishen Tsai, Yasuyuki Matsuda, Evan Chang-Siu, Sumio Sugita, Qingxing Zheng, Xu Chen, Kan Kanjanapas, Dr. Benjamine Fine, Dr. Sandipan Mishra for their friendship. Without all the members in the lab, the graduate school would not have gone that fast and the life here would not have been so colorful and fun. I want to thank Dr. Kyoungchul Kong for his technical insights and suggestions toward my research and the encouragement and support whenever I was having a hard time.

I want to thank all my friends in Berkeley, Michigan, and Taiwan, especially Christine Fong, Yamei Chen, Ming-Chun Tien, Katharine Chang, Wei-Ting Yen, Akira Saito, Hsing-Hua Huang and Chia-Wei Wu. They have been supporting me through the ups and downs in life whenever, wherever. Thank you so much for being my best cheerleaders and always trusting me and giving me the warm and sweet cares when I needed them. I would like to thank Kogo Wolf, for his care, love, and faith in me.

Lastly, I want to thank my dearest family, Eric Han, Iris Han-Liu, Irving Han, and Gloria Han. Thank you for being my motivation, for always supporting my

dream, and for the unlimited love.

Chapter 1

Introduction

1.1 Background

A robot manipulator is a reprogrammable, multifunctional manipulator which is designed to move materials, parts, tools, or specialized devices through variable programmed motions [16]. The robot manipulator (e.g. FANUC M-16iB robot) can be divided into two sections: (1) an arm-and-body, which consists of three joints connected by large links, and (2) a wrist, consisting of two or three compact joints. Attached to the wrist can be a gripper to grasp a work part or a tool (e.g., a spot-welding gun) to perform operations. The two manipulator sections have different functions: the arm-and-body is used to move and position parts or tools in the robot's work space, while the wrist is used to orient the parts or tools at the work location. The robot manipulator is composed of a sequence of link and joint combinations. The actuator at a joint and a link makes a drive train. One drive train may be connected to another drive train or an end-effecter. Drive trains are direct if the actuator is directly coupled to the link (load). Similarly, those that use gear transmission mechanisms are called indirect drive trains.

Introduction of low speed high torque motors has made direct drive robots possible. The biggest advantage of using direct drive trains in robots is the elimination of gear reduction mechanisms, which makes the system much more rigid and thus more desirable from the motion control point of view. Also, joint friction caused mostly by transmission mechanisms can be eliminated. However, without gear reduction, a direct drive motor needs to provide N times large torque than an indirect drive motor, given the same payload. Given the present actuator technology, direct drive motors can be massive and their locations must be carefully selected. For a multi-link manipulator, the increase in mass of links farther away from the base of the robot generally results in a significant increase in required link rigidity. This is usually accompanied by increase in the size and mass of the arm-and-body links.

Indirect drive trains, different from direct drive trains, use gear transmission

mechanisms for the purpose of speed reduction and torque amplification. Harmonic drives and rotor vector (RV) reducers are popular choices for the mechanism. They permit the use of low-torque high-speed motors, which are much smaller in size than direct drive motors. Relatively small motors are adequate because the gears reduce velocities and amplify torques. Moreover, the reducer is designed such that several teeth are engaged at any given time hence making backlash negligible. As a result, they are ideal in applications requiring precision positioning such as semiconductor manufacturing industries.

An indirect drive train, using a harmonic drive/RV reducer in particular, in spite of the advantages as stated above, has several disadvantages. The reducers in indirect drive trains introduce compliance, and nonlinear properties such as friction and hysteresis to the system. Moreover, the servo controllers of indirect drive robot manipulators in most industrial applications utilize position measurement on the motor side, while the system performance is evaluated by the load side information (e.g. the position of the end-effector). Since good motor side performance does not necessarily mean good load side (end-effector) performance, load side sensing information becomes essential to improve the load side performance in indirect drive train systems.

Precise load side position measurements are usually not available for industrial robots due to cost and assembly issues. Although the use of vision sensors is a promising choice to obtain the end-effector position information, the slow sampling rate of the vision system may limit the achievable performance. Thus, low-cost micro-electro-mechanical systems (MEMS) sensors, such as MEMS gyroscopes and accelerometers, are popular choices for motion control applications. Notice that when choosing sensors, the feasibility of sensor installation needs to be considered. The MEMS gyroscopes/accelerometers are small and easy to mount on the load side (end-effector) of robot manipulators without modifying the drive trains.

The ultimate goal of control engineers today is to design high-performance servo systems for these drive trains in a cost effective and timely manner. To achieve this goal and to respond to increasingly stringent performance requirements, the mechatronics approach is more emphasized [59], which considers mechanical hardware, sensing and actuation devices, and servo software as a whole. In this dissertation, the load side acceleration information is used extensively in the control algorithms to improve efficiently and effectively the performance of robot manipulators .

The remainder of this chapter is organized as follows. Section 1.2 states the motivation and contribution of this dissertation, and Section 1.3 provides the outline of the dissertation.

1.2 Motivation and Contribution

1.2.1 Sensor-based Controller Tuning of Indirect Drive Trains

Robot dynamics can vary substantially from one configuration to another. While it is possible to design a controller that stabilizes a robot over the whole workspace, it is very difficult to find a controller that guarantees good performance at every configuration in the presence of system uncertainties. Fortunately, in most of industrial applications the desired trajectories are normally known in advance, and robots are programmed to follow the trajectories repeatedly. Hence, it makes sense to tune controllers specially for the given trajectories. The current practice is to tune the controller gains manually. An experienced control engineer is required to tune/optimize the controller gains for the given trajectories. Since the manual tuning of controllers is very time-consuming, it is desirable to automate the controller gain tuning process.

Various automated tuning methods have been studied for this purpose. The unfalsified control [55] is a tuning method that uses input-output responses of a feedback system excited by sinusoidal inputs at many frequencies in order to select appropriate controller parameters among a set of predefined parameter candidates. The iterative feedback tuning (IFT) is a gain tuning method that updates the controller gains to minimize a certain cost function iteratively [31]. Extremum seeking control (ESC) was used in [37] to tune proportional-integral-derivative (PID) controllers by minimizing a cost function that characterizes the desired behavior of the closed-loop system. The development of automatic gain tuning methods not only reduce the time and cost in robot industries but also enables end users to specify different trajectories for their own specific applications without the presence of a control engineer. This provides them with more flexibility in utilizing the robot.

In this dissertation, a real-time nonlinear programming (RTNLP) method is applied to an indirect drive system for the tuning of feedback and feedforward controllers. The RTNLP method tunes controller gains by minimizing any convex cost function that characterizes the desired behavior of the closed-loop system. Therefore, for the robot system including flexibilities introduced by the harmonic reducer, both the load side and motor side information can be included in the cost function to reflect the desired performance attributes without extra effort. The controller tuning process by the RTNLP method is completely automated. The feedback, feedforward and state feedback controller gains are tuned and compared with the iterative feedback tuning method. The improvement of the load side performance using the tuned controller parameters is also demonstrated experimentally.

1.2.2 Suppression of Oscillations for Indirect drive Robots using Joint Sensing Information

The performance requirements for industrial robots are becoming more and more stringent. Modern robot manipulators are operated under high speed and high accuracy conditions to increase productivity and quality. When robots are driven in a high speed manner, oscillations and vibrations often occur. Such oscillations and vibrations usually depend on the direction and speed of robot motion, and are mostly caused by transmission mechanisms, such as harmonic drives, used in each joint axis of the robot. Although the harmonic drive gear offers many advantages such as compact design, low weight, high speed reduction ratios, and near-zero backlash, it introduces nonlinear characteristics including transmission error, flexibility, and hysteresis [50, 61] into the drive trains.

Among the nonlinear characteristics mentioned above, transmission error is of foremost concern for precision positioning applications. The transmission error is defined as the deviation between the expected output position and the actual output position of a harmonic drive. It introduces an inaccuracy into the kinematics chain and thus causes a positioning error on the output of the gear. In many applications, the positioning error itself is not significant, but its influence on speed variation is critical. A transmission error causes a speed ripple on the gear output shaft, even at a constant rotation speed of the input shaft. These vibrations become dominant when the frequency of the error coincides with the resonant frequency of the control system. Speed fluctuations are largely amplified in such cases and often exceed allowable levels for vibrations[60, 69]. Thus reducing transmission error is important for precision tracking and performance.

To compensate for the transmission error, several control methods have been proposed in literature [18, 20, 28]. Gandhi and Ghorbel [18] proposed nonlinear control algorithms for the compensation of the transmission error in set point and trajectory tracking with harmonic drives. Hirabayashi et al. [28] proposed a method where the controller senses speed ripples through a high resolution encoder and modifies the speed command to the driving motor. Godler et al. [20] applied repetitive control for reducing speed ripples in a harmonic drive system.

Among the different approaches, motor side information (e.g., the motor angle measured by an encoder) is widely used for the feedback control loop. These methods are shown to be effective to some degree. Velocity fluctuation due to the transmission error, however, is most significant on the load side and feedback control cannot effectively reduce the vibrations on the load side by using only motor side information, since it is often unobservable from the motor side. It is important to develop control algorithms that utilize information on the load side to achieve smooth robot motion free from oscillations.

In this dissertation, an adaptive cancellation algorithm is presented to suppress

the vibration phenomena caused by the transmission error of harmonic drives. The transmission error effect is modeled and an analysis scheme that considers the transmission error effect as a disturbance input to the system is constructed. To effectively deal with the oscillations on the load side, load side acceleration information is used in the adaptation laws. Two modifications are proposed to enhance the performance of the algorithm. The effectiveness of the proposed methods, especially on the load side performance, is demonstrated by experiments.

1.2.3 Iterative Learning Control for Indirect Drive Trains

In industrial applications, robots normally follow a trajectory repeatedly. If the repeatability of the robot is good, the trajectory tracking error will become repetitive from one run to another. Iterative learning control (ILC) has been extensively used in control of systems that execute the same task repeatedly. It is a well-established method of control for repetitive processes. Some examples of applying iterative learning control on robot manipulators can be found in [1, 33, 43].

In many learning control applications, the sensor measurement used in the learning controller is the same as that used in the stabilizing feedback controller. For example, in applying iterative learning control to drive trains, the conventional approach [1, 36, 49] is to utilize the motor side measurements neglecting any effects caused by the gear between the motor and the load. In reality, however, there are always mechanical flexibilities in both joints and links. Moreover, the performance of the drive train system is often evaluated by the load side position. If the load side sensor information is available, it is desirable to use load side measurements in the learning controller to directly compensate for the effects of the gear nonlinearities and other disturbances to improve load side tracking performance. Different iterative learning control ideas [43, 64] applied to flexible mechanical systems have been studied previously assuming that the load side position can be measured. The precise load side position measurement, however, is often not available. Although vision sensors are often used to provide precise load side (end-effector) position, achievable performance by using vision sensors may be limited due to the slow sampling rate of the vision system. Notice that iterative learning control is in general considered to be an approach for the purpose of trajectory tracking in the literature [4, 45]. In this dissertation, the learning controller design problem that emphasizes on the purpose of disturbance rejection using the load side acceleration information is presented.

In this dissertation, an optimization-based iterative learning controller scheme for the purpose of disturbance rejection is proposed. This is a systematic and model-based design method, where the trade-off between performance and robustness can be handled. Two iterative learning controllers based on different sensor information are designed and compared. A load side position estimation algorithm based on

Kalman filtering using load side acceleration and motor side position is proposed to provide the load side position estimates in the load side learning controller. The convergence property and the benefits of applying learning controllers for rejecting load side vibrations are confirmed by experiments.

1.2.4 Identification of Dynamics of a Robot Manipulator

Experiments in Section 1.2.1 $\sim$ 1.2.3 are performed on a single-joint indirect drive test stand. The ultimate goal of this research is to generalize the algorithms tested for the single-input-single-output (SISO) test stand to the multi-input-multi-output (MIMO) industrial robots (e.g. FANUC M-16iB robot). An accurate model of the robot dynamics is an essential prerequisite for this goal. An accurate model obtained from system identification is important not only for the controller design, but also for the development of better or new functionalities, such as collision detection, sensorless force control for assembly, and sensorless lead through teaching. System identification also provides an opportunity to gauge the hardware and software capability prior to controller implementation. Therefore, it is important to perform system identification to obtain the estimates for the system parameters.

Robot manipulators, however, are highly nonlinear, coupled multivariable control systems. Unfortunately, the typical SISO system identification approach does not incorporate this coupling effect. Consequently, SISO system identification if often not sufficient to capture the complex dynamics of robot manipulators. As a result, complementary methods such as gain scheduling method, automatic gain tuning method, and adaptive control method are usually applied to improve system performance. If the multi-dimensional system model is identified precisely, it is expected that a multivariable controller can be designed to give better performance over the whole workspace. In addition, it will enable more realistic simulations to evaluate control performance before experiments.

In this dissertation, several aspects of system identification of the FANUC M-16iB robot are presented. First, the SISO identification approach is explained. For the SISO system identification, both closed and open loop techniques are used. The experiments are performed and the results along with the data processing procedure are discussed. The identified parameter values and frequency characteristics are documented so that they can be used in the simulation and servo controller design of the robot. Next, the approach, experiment and data processing procedure for the MIMO system identification are proposed. Even though only preliminary MIMO system identification results are shown, the results demonstrate the effectiveness of the identification approach and can be continued in future research.

1.3 Outline

The remainder of this dissertation is organized as follows. Chapter 2 introduces the model and characteristics of the indirect drive train system. The single-joint indirect drive experimental setup available in the Mechanical Systems Control Laboratory at the University of California, Berkeley is also introduced. In Chapter 3, the sensor based controller gain tuning method using real-time nonlinear programming is presented. The method is used to tune the feedforward, feedback and state feedback controller gains. The algorithm is shown to be not only an online, model-free iterative tuning method, but also capable of optimizing the controller parameters for nonlinear systems with multivariable nonlinear controllers. In Chapter 4, an adaptive disturbance cancellation scheme and its modifications are proposed and utilized to suppress the oscillations caused by transmission error. The load side acceleration feedback is used in the adaptation algorithms in order to further minimize the load side vibrations. Chapter 5 presents the design of the iterative learning controller particularly for suppressing load side vibrations in indirect drive trains. A load side position estimation algorithm is proposed based on the load side acceleration and motor side encoder information. Chapter 6 first discusses in detail the SISO system identification of robot dynamics and results. The discussion then extends to MIMO system identification and the MIMO system identification results for the robot at zero position is presented. Finally, conclusions are provided in Chapter 7

Chapter 2

Indirect Drive Train System

2.1 Introduction

From the point of view of the gear reduction mechanism, robots that use an actuator that is directly coupled to the load (link) are called direct drive robots. On the other hand, those that use gear reduction mechanisms are called indirect drive robots. The use of a compact gear reduction mechanism results in a robot with a relatively small actuation unit that can drive a relatively large payload. The harmonic drive and rotary vector (RV) reducer are two popular choices for the gear reduction unit. They are compact and may achieve a very high gear reduction ratio. This provides several advantages compared to other indirect drive transmission mechanisms such as planetary and spur gears. An indirect drive robot, however, has several disadvantages such as joint compliance, actuator saturation, and harmonic drive nonlinearities. These effects bring challenges to the servo controller design of the indirect drive robot. This chapter describes the experimental setup and the model of a single-joint indirect drive unit used for analysis, controller design, and experiments in the following chapters. Section 2.2 explains the experimental setup and Section 2.3 presents the mathematical model of the single-joint indirect drive train. The indirect drive train characteristics are provided in Section 2.4.

2.2 Single-Joint Indirect Drive Train Experimental Setup

The indirect drive train experimental setup available in the Mechanical Systems Control Laboratory at the University of California, Berkeley is shown in Figure 2.1. This single joint setup consists of: 1) a servo motor with a $20,000$ counts/revolution encoder, 2) a harmonic drive with a 80:1 gear ratio, 3) a load-side $144,000$ counts /revolution encoder, 4) and a payload. Notice that if the link as shown in the figure

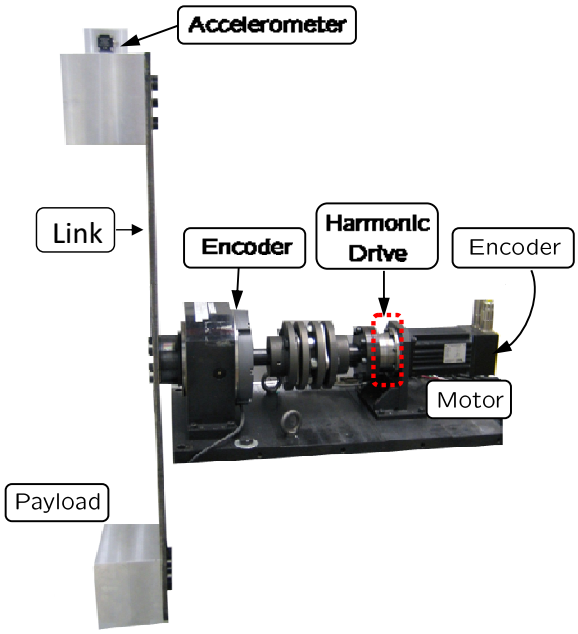


Figure 2.1: Single-joint indirect drive train experimental setup

is rigid, the load side rotary encoder provides perfect knowledge of the payload position. Since in most cases the payload position is not available, it is assumed that the load side rotary encoder measures the payload position.

Besides the encoder, two accelerometers (Kistler, Type: 8330A3) are installed at the ends of the payload symmetrically as shown in Figure 2.2. The accelerometers are arranged to compensate for the effects of gravity on the accelerometer measurements. This configuration has the following relationship from the linear acceleration measurements, a_{cc1} and a_{cc2}, to the load side angular acceleration, $\ddot{\theta}_\ell$, and the gravity, g

$$\begin{aligned} a_1 &= R_a\ddot{\theta}_\ell + g\cos\bar{\theta}_\ell \\ a_2 &= R_a\ddot{\theta}_\ell - g\cos\bar{\theta}_\ell \end{aligned} \tag{2.1}$$

where R_a is the distance from the rotation axis to the accelerometer location, and $\bar{\theta}_\ell$ is the angle of the payload from the horizontal position. It is not necessary to measure the angle $\bar{\theta}_\ell$, since the angular acceleration $\ddot{\theta}_\ell$ can be obtained as

$$\ddot{\theta}_\ell = \frac{a_1 + a_2}{2R_a} \tag{2.2}$$

which is independent of any gravity effects.

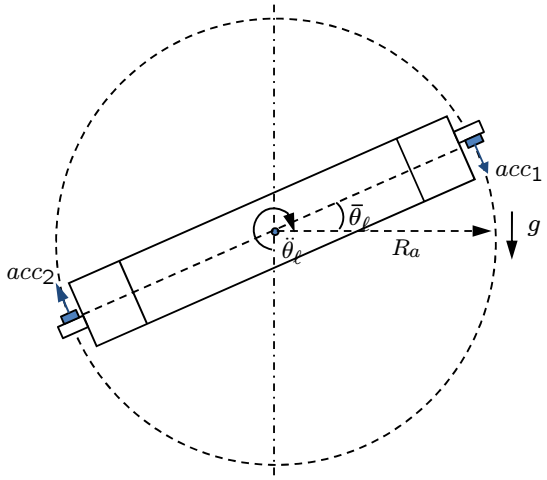


Figure 2.2: Accelerometer installation [68]

The controller for research is implemented in a LabVIEW real-time target installed with LabVIEW real-time and FPGA modules. The sampling rate is selected to be 1 kHz for all the implementation of studies on this setup.

2.3 Single-Joint Indirect Drive Model

The schematic of the single-joint indirect drive train for the experimental setup in Figure 2.1 is shown in Figure 2.3. In the figure, J_m and J_ℓ represent the motor side and the load side inertia, respectively. d_m, d_j and d_ℓ are respectively the motor side, the joint, and the load side damping. k_j is the coefficient of joint stiffness. θ_m and θ_ℓ represent the motor side and load side position, respectively. N is the gear reduction ratio, and u is the control input which is the motor torque in this case. f_m, f_ℓ, and f_h represent the nonlinear friction forces at the motor side, the load side, and the reducer, respectively.

Note that while the actual robot joint is inherently nonlinear, a good linear approximation can still preserve satisfactory performance if the nonlinear elements are negligible or the linear parts are of most interest. Additionally, linear model allows the use of various linear analysis and controller synthesis methods. The linear

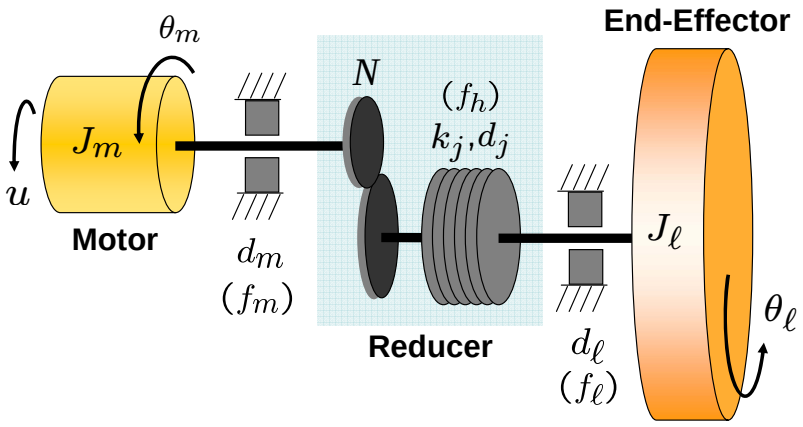


Figure 2.3: Single-joint indirect drive train

model of the indirect drive train in Figure 2.3 can be represented as

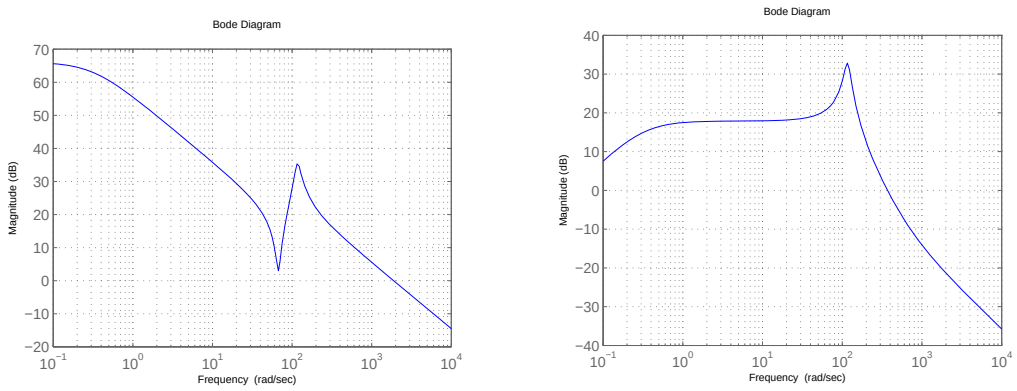
$$\begin{aligned} J_m\ddot{\theta}_m + d_m\dot{\theta}_m &= -\frac{1}{N}\left[k_j\left(\frac{\theta_m}{N} - \theta_\ell\right) + d_j\left(\frac{\dot{\theta}_m}{N} - \dot{\theta}_\ell\right)\right] + u \qquad (2.3) \\ J_\ell\ddot{\theta}_\ell + d_\ell\dot{\theta}_\ell &= k_j\left(\frac{\theta_m}{N} - \theta_\ell\right) + d_j\left(\frac{\dot{\theta}_m}{N} - \dot{\theta}_\ell\right) \end{aligned}$$

Note that θ_m and θ_ℓ are coupled in both equations. The state-space formulation of (2.3) can be written as

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \qquad (2.4)$$

where

$$\begin{aligned} x &= [\theta_m \;\; \dot{\theta}_m \;\; \theta_\ell \;\; \dot{\theta}_\ell]^{\mathrm{T}} \\ \mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_j}{N^2 J_m} & -\frac{d_m + d_j/N^2}{J_m} & \frac{k_j}{NJ_m} & \frac{d_j}{NJ_m} \\ 0 & 0 & 0 & 1 \\ \frac{k_j}{NJ_\ell} & \frac{d_j}{NJ_\ell} & \frac{-k_j}{J_\ell} & -\frac{d_j + d_\ell}{J_\ell} \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 0 & \frac{1}{J_m} & 0 & 0 \end{bmatrix}^{\mathrm{T}} \end{aligned}$$



(a) $\frac{\dot{\theta}_m}{u}$ (b) $\frac{\ddot{\theta}_\ell}{u}$

Figure 2.4: Magnitude of the frequency response of the single-joint indirect drive setup

2.4 Model Characteristics

From (2.3), the transfer functions from the torque input to the motor side velocity and the load side acceleration can be found as:

$$\frac{\dot{\theta}_m(s)}{u(s)} = \frac{J_\ell s^2 + (d_j + d_\ell)\, s + k_j}{J_m J_\ell s^3 + J_d s^2 + J_k s + k_j \left(d_m + \frac{d_\ell}{N^2}\right)} \tag{2.5}$$

$$\frac{\ddot{\theta}_\ell(s)}{u(s)} = \frac{d_j s^2 + k_j s}{N\left[J_m J_\ell s^3 + J_d s^2 + J_k s + k_j \left(d_m + \frac{d_\ell}{N^2}\right)\right]} \tag{2.6}$$

where

$$J_d = J_m (d_j + d_\ell) + J_\ell \left(\frac{d_j}{N^2} + d_m\right)$$

$$J_k = J_m k_j + \frac{J_\ell k_j}{N^2} + (d_j + d_\ell)\, d_m + \frac{d_j d_\ell}{N^2}$$

Figure 2.4(a) and 2.4(b) show the frequency responses of (2.5) and (2.6), respectively. Note that the system identification of the experimental setup in Figure 2.1 was performed previously and the identified system parameters are shown in Table 2.1. It can be seen from Figure 2.4(a) that the anti-resonant and resonant frequencies of the experimental setup are approximately 11Hz and 19Hz, respectively.

Table 2.1: Identified parameters for the single-joint robot

Variable	Value	Units
Motor side inertia , J_m	5.313×10^{-4}	Nms^2
Load side inertia, J_ℓ	6.8	Nms^2
Joint stiffness, k_j	31000	Nm/rad
Joint damping, d_j	47	Nms
Motor side damping, d_m	5×10^{-4}	Nms
Load side damping, d_ℓ	0	Nms
Gear ratio, N	80	-

Friction effect

Control of indirect drive train is challenging in servo control problems not only due to joint compliance but also nonlinearities such as friction and torque hysteresis. In the system shown in Figure 2.3, the energy is dissipated mainly by three friction forces: the motor bearing friction, f_m, the load output bearing friction, f_ℓ, and the gear meshing friction, f_h. In most cases, the gear meshing friction f_h is dominant [57] and the load side friction, f_ℓ, which is often neglected, is smaller compared to the motor side one. Limit cycles, however, may be caused by load side Coulomb friction in standard industrial robots [39]. Furthermore, the load side friction can be more problematic in the sense that the control effort from the motor can only be transmitted through the flexible joint mechanism. This makes the compensation of the load side friction more difficult than that of the motor side friction. It is important to consider all three friction forces while modeling the system with the consideration of friction.

The combination of system model with friction effect can be written as [9]

$$\begin{aligned} J_m\ddot{\theta}_m &= -\frac{1}{N}\left[k_j\left(\frac{\theta_m}{N}-\theta_\ell\right)+d_j\left(\frac{\dot{\theta}_m}{N}-\dot{\theta}_\ell\right)\right]+u-(f_m+f_h) \qquad (2.7)\\ J_\ell\ddot{\theta}_\ell &= k_j\left(\frac{\theta_m}{N}-\theta_\ell\right)+d_j\left(\frac{\dot{\theta}_m}{N}-\dot{\theta}_\ell\right)-f_\ell \end{aligned}$$

Equation (2.7) gives

$$\begin{aligned} J_m\ddot{\theta}_m+\frac{1}{N}J_\ell\ddot{\theta}_\ell &= u-F \qquad (2.8)\\ F &= f_m+f_h+\frac{f_\ell}{N} \end{aligned}$$

To describe the friction effects, the Lund-Grenoble (LuGre) model [8] is often used. For the single-joint indirect drive setup, the static friction can be modeled by

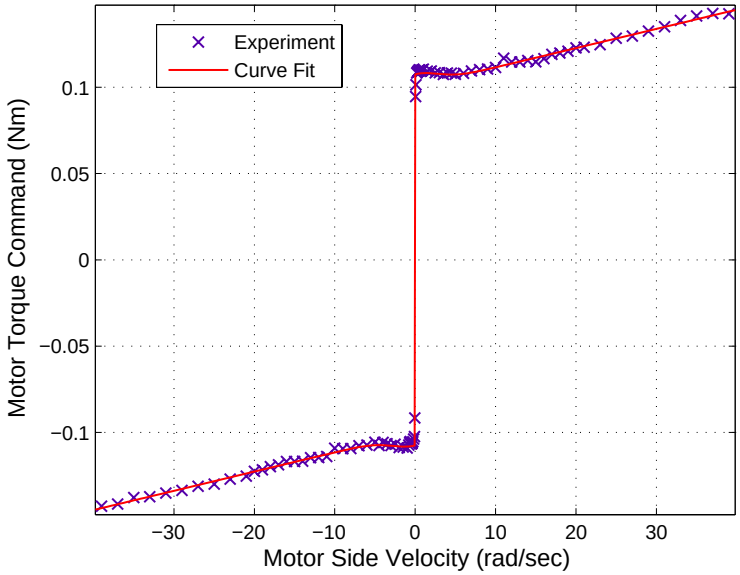


Figure 2.5: Static friction identification result

Table 2.2: The identified friction parameters (SI Units)

F_c	F_s	v_s	σ_2
0.1004	0.1075	3.951	0.001114

simplifying the LuGre model as

$$F\left(\dot{\theta}_m\right) = \left[F_c + (F_s - F_c)\, e^{\left(-\dot{\theta}_m^2/v_s^2\right)}\right] \operatorname{sgn}\left(\dot{\theta}_m\right) + \sigma_2 \dot{\theta}_m \tag{2.9}$$

where σ_2 represents the macro-damping of the velocity $\dot{\theta}_m$. F_C and F_S are the levels of Coulomb friction and stiction forces. v_s is the Stribeck velocity. The static velocity-torque (friction force) characteristic for the single-joint experimental setup is shown in Figure 2.5.

Each point in Figure 2.5 is obtained by keeping the motor side velocity constant for the same amount of distance in a closed-loop manner. The experiment is repeated at various velocities for the same path to obtain the whole static velocity-torque map. The nonlinear least squares method in the Optimization Toolbox of MATLAB is applied to obtain the static friction parameters, F_C, F_S, v_s, and σ_2. The identified values are listed in Table 2.2.

2.4.1 Summary

In this chapter, a detailed description of the single-joint indirect drive train experimental setup was presented. The two-inertia mathematical model for the experimental setup was then constructed and the model characteristic including the friction effect was discussed. The single-joint indirect drive model is used extensively in this dissertation for controller design and analysis purposes. Moreover, the single-joint indirect drive setup is used for the experimental verification of the proposed control algorithms throughout the dissertation.

Chapter 3

Sensor-based Controller Tuning of Robot Manipulators by Real-time Optimization

3.1 Introduction

In industrial motion control systems, the structure of control algorithms is often fixed, and thus the performance is optimized by tuning the controller gains. The manual tuning of controllers is a time-consuming task especially when it is required to optimize many parameters. To address this difficulty, much effort has been invested in developing systematic tuning methods. Optimal control theories such as linear quadratic control and H_∞ control usually provide appropriate controller gains for a given system and a cost function. In [14], the H_∞ approach was used to determine the controller gains. The model parameters and the uncertainty bounds are identified by the dual Youla parameterization, and the controller gains are determined based on μ-synthesis. This method is attractive due to guaranteed robustness. It is, however, computationally demanding, and the performance in practice may not be necessarily optimal, since it only deals with the worst case situation. Moreover, the theoretical values of controller gains obtained from optimal control theories may not result in the desired performance in practice due to unmodeled and/or uncertain properties of actual systems. In such cases, the controller gains need to be further tuned by experiments to refine the performance.

Various model-free experimental tuning methods have been studied to obtain appropriate controller gains [30, 37, 55]. The unfalsified control [55] is a tuning method that uses input-output responses of a feedback system excited by sinusoidal inputs for many frequencies to select appropriate controller parameters among a set of predefined parameter candidates. An adaptive algorithm is used to update the controller parameters when the controller falsifies a given criterion. Unfalsified

control for an infinite set of proportional-integral-derivative (PID) controllers has been developed. This approach requires a carefully chosen input signal, which is a practical difficulty.

Another example of model-free controller tuning method is iterative feedback tuning (IFT) [30, 32, 68]. The IFT method optimizes the controller gains based on the experimental data and updates the controller gains to minimize a certain cost function iteratively in a closed-loop setting [31]. It has been applied to various applications [22, 24, 68]. In [24] and [68], the IFT method was applied to a two-mass-spring system. In [24], it is assumed that the load-side position measurement is available for feedback control, and in [68], the load-side acceleration information was used for tuning of the motor loop feedback controller.

The IFT method is particularly attractive when the system is linear and the performance index is quadratic. In such cases, the gradient estimate of the performance index, which is required to optimize the parameters can be obtained by calculations using data from a set of additional experiments, i.e. gradient experiments. After the gradient estimate is computed, the controller gains are updated. Although the IFT technique can be applied to multi-variable non-linear systems, the derivation and design of the gradient experiments may be very complicated.

In this chapter, the controller gains are optimized based on real-time nonlinear programming (RTNLP) by perturbing the controller gains [40]. The method estimates the gradient by perturbation in real-time. Since the gradient estimate is obtained while the controller gains are updated, it does not require the design or the execution of any additional experiment for obtaining the gradient. Moreover, the cost function can be an arbitrary convex function. Notice that any non-convex function can be converted into the convex form by utilizing certain functions, such as barrier functions [2]. Therefore, the RTNLP algorithm is not only an on-line, model-free iterative tuning method, but it is also capable of optimizing the controller parameters for non-linear systems with multi-variable non-linear controllers. In this chapter, the RTNLP method is implemented in the discrete-time domain on the single-joint indirect drive train unit described in Section 2.2. To deal with the joint compliance in indirect drive systems, the load side sensor information as well as the motor side information are included in the performance index during the tuning process to achieve better performance on the load side. More specifically, the load side acceleration information is utilized because the MEMS accelerometers nowadays are small and easy to mount and the acceleration signal directly reflects the vibrations on the load side.

The remainder of this chapter is organized as follows. Section 3.2 provides the model and the controller structure of the single-joint indirect drive train system. The controller tuning methodology based on the sensor measurements is presented in Section 3.3 and the experimental results are shown in Section 3.4. The tuning of the state feedback controller is presented in Section 3.5 and finally conclusions are

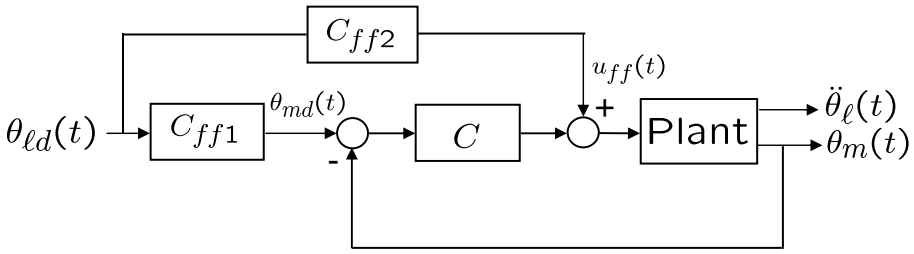


Figure 3.1: Block diagram of the control system

given in Section 3.6.

3.2 Single-Joint Indirect Drive Train System

3.2.1 Model

The dynamic equations of the single-joint indirect drive train were expressed by (2.3)

$$\begin{aligned} J_\ell \ddot{\theta}_\ell + d_\ell \dot{\theta}_\ell &= k_j \left(\frac{\theta_m}{N} - \theta_\ell \right) + d_j \left(\frac{\dot{\theta}_m}{N} - \dot{\theta}_\ell \right) \\ J_m \ddot{\theta}_m + d_m \dot{\theta}_m &= u - \frac{1}{N} \left[k_j \left(\frac{\theta_m}{N} - \theta_\ell \right) + d_j \left(\frac{\dot{\theta}_m}{N} - \dot{\theta}_\ell \right) \right] \end{aligned}$$

3.2.2 Controller Structure

Figure 3.1 shows the servo controller structure for the robot system (2.3). It consists of two feedforward controllers, C_{ff1} and C_{ff2}, and a motor position feedback controller, C. θ_{md} and $\theta_{\ell d}$ represent the desired motor side and load side trajectory, respectively.

Assuming that the joint damping, d_j, is negligible in (2.3), the desired motor side position, $\theta_{md}(t)$, and the control input, $u(t)$, can be determined such that the system dynamics is compensated, i.e.

$$\theta_{md}(t) = N\theta_{\ell d}(t) + \frac{N}{k_j}(J_\ell \ddot{\theta}_{\ell d}(t) + d_\ell \dot{\theta}_{\ell d}(t)) \tag{3.1}$$

$$u_{ff}(t) = J_m \ddot{\theta}_{md}(t) + d_m \dot{\theta}_{md}(t) + \frac{1}{N}(J_\ell \ddot{\theta}_{\ell d}(t) + d_\ell \dot{\theta}_{\ell d}(t)) \tag{3.2}$$

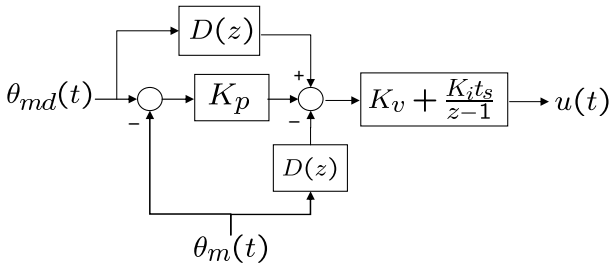


Figure 3.2: Block diagram of the feedback controller

Notice that (3.3) and (3.2) result in two feedforward controllers, i.e.

$$\begin{aligned} C_{ff1}(s) &= \frac{N}{k_j}[k_j + d_\ell s + J_\ell s^2] \\ &= N + K_{f1}s + K_{f2}s^2 \qquad (3.3) \\ C_{ff2}(s) &= \frac{N}{k_j}[(k_j d_m + \frac{k_j}{N^2}d_\ell)s + (J_m k_j + d_m d_\ell + \frac{k_j}{N^2}J_\ell)s^2 \\ &\quad +(J_\ell d_m + J_m d_\ell)s^3 + J_m J_\ell s^4] \\ &= K_{f3}s + K_{f4}s^2 + K_{f5}s^3 + K_{f6}s^4 \qquad (3.4) \end{aligned}$$

where $C_{ff1}(s)$ and $C_{ff2}(s)$ are the transfer functions from the load side reference, $\theta_{\ell d}(t)$, to the motor side trajectory, $\theta_{md}(t)$, and to the control input, $u(t)$, respectively. Note that s^4 term is included in $C_{ff2}(s)$, which requires the fourth derivative of the trajectory. From (3.3), it is observed that the controller parameters in $C_{ff1}(s)$ and $C_{ff2}(s)$, i.e. $[K_{f1}, ..., K_{f6}]$, can be calculated using the identified physical parameters, e.g. $K_{f1} = \frac{N}{k_j}d_\ell$. However, the parameters of the actual system are often difficult to identify precisely. In such cases, the controller gains, $[K_{f1}, \ldots, K_{f6}]$, need to be experimentally tuned for achieving the desired performance in practice. In this chapter, the six feedforward controller parameters are tuned. Note that in the experiments, the feedforward controllers are implemented in discrete time by replacing the operator s with $\frac{z-1}{t_s z}$ where t_s is the sampling period.

The system dynamics is completely compensated by the feedforward controllers in ideal situations. However, a feedback controller is necessary when disturbances or model uncertainties are present. Figure 3.2 shows the structure of the motor position feedback controller, C, which is a modified PID controller. The discrete-time transfer function from the error, $\theta_{md}(t) - \theta_m(t)$, to the control input, $u(t)$, can be written as

$$C(z) = [K_p + D(z)][K_v + K_i I(z)] \qquad (3.5)$$

where K_p, K_v, and K_i are the controller gains. $D(z)$ represents the approximate

differentiation, i.e $D(z) = \frac{z-1}{t_s z}$. $I(z)$ is the discrete time integrator, i.e $I(z) = \frac{t_s}{z-1}$. In this chapter, three feedback controller gains, i.e. K_p, K_v, and K_i, are optimized.

3.3 Controller Tuning Methodology

3.3.1 Cost Functions

Feedback controller tuning

The controller tuning method in this chapter optimizes the controller parameters such that the value of a cost function is minimized. Since the feedback and the feedforward controller can interact with each other in one optimization process, they are optimized separately. That is, the feedforward controllers, C_{ff1} and C_{ff2}, are disabled during the tuning process of the feedback controller gains and the motor side reference trajectory, θ_{md}, is used instead of the load side reference, $\theta_{\ell d}$. To define a cost function, let $\Theta_{fb}(i)$ represent the collection of the feedback controller gains at the i^{th} iteration, i.e. $\Theta_{fb}(i) = [K_p(i), K_v(i), K_i(i)]$. Then, the gains are adjusted in each iteration to minimize the performance index

$$J_{fb}(\Theta_{fb}(i)) = \sum_{k=k_0}^{k_f} [M_b(k)e_m^2(k) + U_b(k)u^2(k) + A_b(k)e_a^2(k)] \tag{3.6}$$

where $e_a(k) = \ddot{\theta}_{\ell d}(k) - \ddot{\theta}_{\ell}(k)$, $e_m(k) = \theta_{md}(k) - \theta_m(k)$, and $u(k)$ is the control input. k is the time index during one trial period, and i represents the iteration index. The cost function in (3.6) penalizes the motor position error, the control input, and the load side acceleration error over the interval $k = [k_0, k_f]$ where the load-side acceleration error is a measure of vibrations. By setting $[k_0,\ k_f] = [1,\ N_f]$ where N_f is the duration of each trial, (3.6) penalizes the three terms over the entire time during one iteration. The trade-off among the penalty terms is determined by the weighting factors, $M_b(k)$, $U_b(k)$, and $A_b(k)$.

Feedforward controller tuning

Without loss of generality, assume that the feedback controller, $C(z)$, is tuned before the tuning of the feedforward controllers. Given a desired load side positoin reference, $\theta_{\ell d}(k)$, the feedforward controllers are tuned to reduce the load side tracking error, $e_\ell(k) = \theta_{\ell d}(k) - \theta_\ell(k)$. Since the load-side position measurement is not available, the feedforward controllers are adjusted to minimize the load-side acceleration error, $e_a(k)$, and the motor side position error, $e_m(k)$. It follows that the

performance index for the feedforward controller tuning is selected as

$$J_{ff}(\Theta_{ff}(i)) = \sum_{k=k_0}^{k_f} [M_f(k)e_m^2(k) + A_f(k)e_a^2(k)] \tag{3.7}$$

where $\Theta_{ff}(i) = [K_{f1}(i), \ldots, K_{f6}(i)]$. $M_f(k)$ and $A_f(k)$ are the weighting factors that determine the ratio between the two penalty terms.

3.3.2 Update Law

The cost functions, $J_{fb}(\Theta_{fb}(i))$ and $J_{ff}(\Theta_{ff}(i))$, can be seen as a mapping from the control parameters to the measure of performance. The real-time optimization method tunes the controller gains based on the gradient of the cost function. The controller gains are updated iteratively using the conjugate gradient method, i.e.,

$$\Theta(i+1) = \Theta(i) - \gamma_i D(i) \tag{3.8}$$

where

$$\begin{aligned} D(i) &= -\nabla J(\Theta(i)) + \beta_i D(i-1), \\ \beta_i &= (\|\nabla J(\Theta(i))\| / \|\nabla J(\Theta(i-1))\|)^2, \\ D(0) &= -\nabla J(\Theta(0)), \end{aligned}$$

$J(\Theta(i))$ is the cost function and $\Theta(i) \in \Re^q$ represents the controller gain vector to be optimized. where q is the number of variables to be optimized. $\gamma_i \in \Re_+$determines the update speed, and $\nabla J(\Theta(i))$ is the gradient of the cost function. $D(i)$ is the searching direction. The conjugate gradient method [2] is simple but effective. It is known that in the method, the current searching direction is not orthogonal to the previous direction and thus it tends to cut diagonally through the orthogonal steepest descent directions. Therefore, it improves the rate of convergence of the steepest descent method considerably. Notice that the first step of the conjugate gradient method is the same as the steepest descent method. Each cost function in (3.6) and (3.7) is a convex function that includes three and six variables, respectively. Equation (3.8) updates the variables such that the value of the cost function is decreased.

3.3.3 Estimation of ∇J in Real-Time

The gradient is numerically estimated by amplitude modulation. Suppose $\Theta(i)$ is perturbed by $\varepsilon(i)$. The Taylor series expansion of $J(\Theta(i) + \varepsilon(i))$ is

$$\begin{aligned} J(\Theta(i) + \varepsilon(i)) &= J(\Theta(i)) + \varepsilon(i)^T \nabla J(\Theta(i)) \\ &\quad + 0.5\varepsilon(i)^T \mathbf{H}(\Theta(i))\varepsilon(i) + E(\varepsilon(i)^3) \end{aligned} \tag{3.9}$$

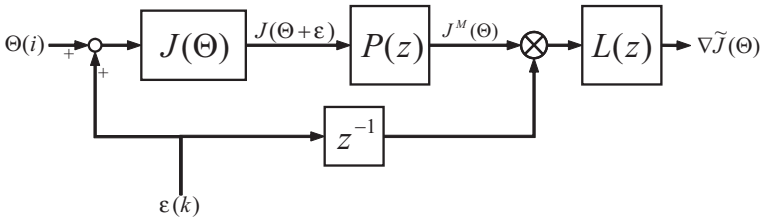


Figure 3.3: Block diagram for gradient estimation

$\nabla J(\Theta(i))$ and $H(\Theta(i))$ are the gradient and the Hessian matrix of $J(\Theta(i))$, respectively. $E(\varepsilon^3(i))$ is the higher order terms and is neglected here. The perturbation, $\varepsilon(i)$, is a vector that consists of q sinusoidal waves, i.e.,

$$\varepsilon(i) = \begin{bmatrix} a_1 \sin(\omega_{c1} i N_f t_s) \\ a_2 \sin(\omega_{c2} i N_f t_s) \\ \vdots \\ a_q \sin(\omega_{cq} i N_f t_s) \end{bmatrix} \in \Re^q \tag{3.10}$$

a_l and ω_{cl} are the amplitude and the frequency of the perturbation for the l^{th} variable, respectively. Notice that frequencies should be selected such that each ω_{cl} is distinguished from each other, and $\omega_{cl} \neq \omega_{cm} + \omega_{cn}$ for $l, m, n = 1, \ldots, q$.

Notice that $\sin(\omega_1) \times \sin(\omega_2)$ has the frequency components of $\omega_1 + \omega_2$ and $|\omega_1 - \omega_2|$. Therefore, in (3.9) the perturbation frequency is preserved only in the first order term, i.e. $\varepsilon(i)^T \nabla J(\Theta(i))$. The second term in (3.9), in which $\nabla J(\Theta(i))$ is included, is extracted by applying a frequency filter that passes only the concerned frequency component. For this purpose, a peak filter is applied, i.e.

$$p_i(s) = \frac{b\omega_{cl} s}{s^2 + b\omega_{cl} s + \omega_{cl}^2} \tag{3.11}$$

Figure 3.4 shows the frequency response of the peak filter with the center frequency, ω_{cl}, at 250Hz. As shown in the figure, b determines the width of the peak. To extract only a certain frequency component, b should be sufficiently small. Notice that $p_i(s)$ introduces no phase delay at the center frequency (see the dashed line in Figure 3.4).

For the implementation purpose, (3.11) needs to be discretized. Since the center frequencies are critical in the design of the peak filters, a method which does not change the locations of poles and zeros should be applied for the peak filters. Note that a pole in the continuous time domain (p) is equivalent to e^{pT} in the discrete time domain. $p_i(s)$ has two poles at $\omega_{ci}[-0.5b \pm (1 - (b/2)^2)^{0.5} i]$ and a zero at 0.

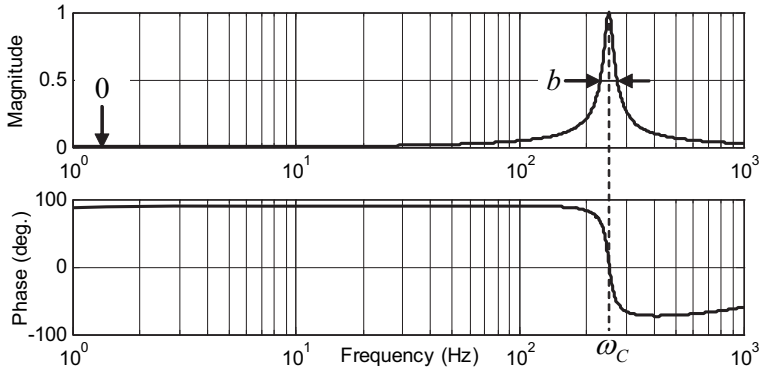


Figure 3.4: Frequency response of a peak filter

Therefore, a discretized peak filter is

$$p_i(z) = \frac{b_0(z-1)}{z^2 + a_1 z + a_0} \tag{3.12}$$

where

$$\begin{aligned} a_0 &= e^{-b\omega_{cl}t_s N_f} \\ a_1 &= -2e^{-0.5b\omega_{cl}t_s N_f}\cos\left(\sqrt{1-(\frac{b}{2})^2}\omega_{cl}t_s N_f\right) \\ b_0 &= \left|\frac{e^{j2\omega_{cl}t_s N_f} + a_1 e^{j\omega_{cl}t_s N_f} + a_0}{e^{j\omega_{cl}t_s N_f} - 1}\right| \end{aligned}$$

In the q dimensional case, a peak filter vector, $P(z)$, is defined as

$$P(z) = [p_1(z) \;\; p_2(z) \;\; \ldots \;\; p_q(z)]^T \tag{3.13}$$

where $p_l(z)$ is the peak filter designed for the perturbation frequency, ω_{cl} in (3.10) for the l^{th} variable. The perturbed cost function filtered by $P(z)$ can be obtained as

$$\begin{aligned} P(z)[J(\Theta(i)+\varepsilon(i))] &= P(z)[J(\Theta(i))] + P(z)[\varepsilon(i)^T \nabla J(\Theta(i))] \\ &\quad + P(z)[0.5\varepsilon(i)^T \mathbf{H}(\Theta(i))\varepsilon(i)] + P(z)[E(\varepsilon(i)^3)] \\ &\approx \mathbf{E}_p(i)\nabla J(\Theta(i)) \\ &\equiv \mathbf{J}^M(\Theta(i)) \in \Re^{q\times 1} \end{aligned} \tag{3.14}$$

Notice that $J^M(\Theta(i))$ is the gradient modulated by $E_P(i)$, where

$$\mathbf{E}_p(i) \equiv \mathrm{diag}[a_1 \sin(\omega_{c1} i N_f t_s), \ldots, a_q \sin(\omega_{cq} i N_f t_s)] \in \Re^{q\times q} \tag{3.15}$$

Equation (3.15) holds only if b in the peak filter is sufficiently small. Notice that the second order term in the perturbed cost function, i.e., $0.5\varepsilon(i)^T\mathbf{H}(\Theta(i))\varepsilon(i)$ in (3.9), is eliminated by applying the peak filter vector in (3.13).

To obtain $\nabla\mathbf{J}(\Theta(i))$ from $J^M(\Theta(i))$, (3.14) needs to be demodulated. In the demodulation process, the perturbation signals, $\varepsilon(i)$, are multiplied to the modulated cost function, $\mathbf{J}^M(\Theta(i))$, i.e.

$$\varepsilon(i)\dot{\otimes}\mathbf{J}^M(\Theta(i)) = \mathbf{E}_p^2(i)\nabla J(\Theta(i)) \in \Re^{q\times 1} \tag{3.16}$$

where $E_P^2(i)$ is a diagonal matrix with the components of $a_l^2\sin^2(\omega_{cl}iN_ft_s)$ and $\dot{\otimes}$ generates a vector of pointwise product, i.e.,

$$\begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} \dot{\otimes} \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \equiv \begin{bmatrix} a_1b_1 \\ \vdots \\ a_nb_n \end{bmatrix} \in \Re^n$$

The demodulated signal is also decoupled due to the diagonality of $\mathbf{E}_P^2(i)$. Using a trigonometric function, $\sin^2(\omega_{cl}iN_ft_s) = 0.5[1-\cos(2\omega_{cl}iN_ft_s)]$, $\mathbf{E}_P^2(i)$ can be divided into two terms, i.e.

$$\mathbf{E}_p^2(i) = \mathbf{A}[\mathbf{I}+\mathbf{B}(i)] \in \Re^{q\times q} \tag{3.17}$$

where

$$\begin{aligned} \mathbf{A} &= 0.5\text{diag}[a_1^2, a_2^2, \ldots, a_q^2] \in \Re^{q\times q} \\ \mathbf{B}(i) &= -\text{diag}[\cos(2\omega_{c1}iN_ft_s), \ldots, \cos(2\omega_{cq}iN_ft_s)] \in \Re^{q\times q} \end{aligned}$$

$\mathbf{A}$ is a constant matrix, and $\mathbf{B}(i)$ has the frequency components related to the perturbation. From (3.16), it is desired to extract $\nabla J(\Theta(i))$, i.e. $\mathbf{B}(i)$ in (3.17) needs to be eliminated. This is solved by applying a lowpass filter to (3.16). Notice that any lowpass filter is applicable to eliminate $\mathbf{B}(i)$ as long as its cut-off frequency is lower than the lowest frequency in $\varepsilon(i)$. Since the demodulated signal is q dimensional, the filter matrix, $\mathbf{L}(z)$, is designed as

$$\mathbf{L}(z) = l(z)\mathbf{A}^{-1} \tag{3.18}$$

where $l(z)$ is a single lowpass filter that satisfies the condition mentioned above. Equation (3.18) can be represented as $\text{diag}[2a_1^{-2}l(z), 2a_2^{-2}l(z), \ldots, 2a_q^{-2}l(z)]$, where a_l are the perturbation amplitudes. The demodulation is accomplished by applying $\mathbf{L}(z)$ to (3.16), i.e.

$$\begin{aligned} \mathbf{L}(z)[\varepsilon(i)\dot{\otimes}\mathbf{J}^M(\Theta(i))] &= l(z)[\mathbf{A}^{-1}\mathbf{A}[\mathbf{I}+\mathbf{B}(i)]\nabla J(\Theta(i))] \\ &\approx \nabla J(\Theta(i)) \end{aligned} \tag{3.19}$$

The stability of the algorithm is guaranteed by the choice of γ, i.e., the step size, in (3.8). The stability proof of the algorithm can be found in [40].

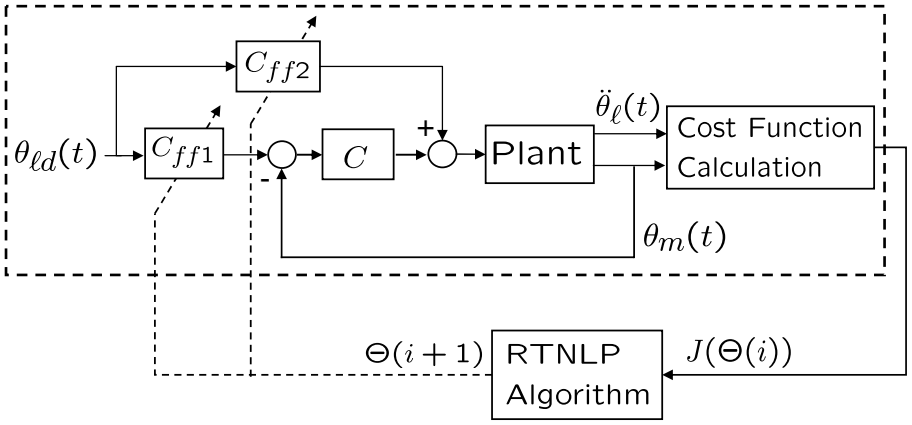


Figure 3.5: The feedforward controller tuning scheme

3.3.4 Controller Tuning Scheme

Figure 3.5 shows the example of the feedforward controller tuning scheme. The experiment shown inside the dashed box is run iteratively; that is, one experiment is run given the controller parameters, $\Theta(i)$, and the value of the cost function, $J(\Theta(i))$, is calculated after the data of one run is acquired. Then, the real-time optimization algorithm uses the value of $J(\Theta(i))$ to update the controller parameters. A new experiment is then performed with the new controller parameters, i.e. $\Theta(i+1)$, and the process continues iteratively. Notice that for the gradient estimation, a perturbation is used in the RTNLP algorithm. For the tuning of one parameter, one perturbation frequency is needed, e.g., for the feedforward controller tuning process, six perturbation frequencies are needed, i.e., $\varepsilon_{ff}(i) = [a_1 \sin(\omega_{c1} i N_f t_s), \cdots, a_6 \sin(\omega_{c6} i N_f t_s)]$. The details of how to choose these frequencies properly were explained in Section 3.3.3. In the following section, the controller tuning method is applied to the indirect drive train system discussed in Section 2.2.

3.4 Controller Tuning Results

3.4.1 Iterative Feedback Tuning (IFT)

The tuning results of the RTNLP method are compared with those of the iterative feedback tuning (IFT) method. The IFT method is a model free technique that tunes the parameters of a fixed structure controller in a closed-loop setting. In IFT, the controller parameters are optimized by using a gradient estimate of the cost function that is calculated based on data collected from closed-loop experiments.

The controller gains can be updated by the iterative algorithm,

$$\Theta(i+1) = \Theta(i) - \gamma_i \mathbf{R}_i^{-1} \nabla J(\Theta(i)) \tag{3.20}$$

where $\Theta(i)$ represents the controller gain vector of the i^{th} iteration, γ_i is a positive real scalar that determines the step size in the i^{th} iteration, and $\mathbf{R}_i$ is an appropriate positive definite matrix. If R_i is an identify matrix, the algorithm follows the rule of the steepest descent method. In the case that $\mathbf{R}_i = \mathbf{H}(\Theta(i))$ or an approximated Hessian, $J(\Theta(i))J^T(\Theta(i))$, (3.20) follows the Newton or Gauss-Newton algorithm where $\mathbf{H}(\Theta(i))$ is the Hessian matrix. The choice of $\mathbf{R}_i$ and γ_i will thus affect the convergence of the IFT method [34]. The step size γ_i for the IFT method used in this chapter is determined using the backtracking line search method [2] and the tuning process is terminated if the step size is smaller than a certain criteria. The estimate of the gradient is obtained from several gradient experiments designed previously and executed after the normal experiment. More detail on the design of gradient experiments for iterative feedback tuning of the single-joint indirect drive robot can be found in [68].

Both IFT and RTNLP methods are non-model based, closed-loop, and online tuning methods that utilize the gradient of the cost function to find the optimal controller parameters. The difference mainly lies in how these algorithms estimate the gradient. In this chapter, the same cost functions are used in both methods for fair comparison. $\mathbf{R}_i$ is chosen to be the approximation of the Hessian, $J(\Theta(i))J^T(\Theta(i))$, where they can be obtained simultaneously with the gradients.

3.4.2 Feedback Controller Tuning

As stated in section 3.3, the feedforward controllers, C_{ff1} and C_{ff2}, are disabled during the feedback controller tuning process. The proposed and the IFT method were used to tune the three feedback controller gains, i.e. $[K_p,\ K_v,\ K_i]$ in (3.5). The initial values of the gains were set to be $[K_p,\ K_v,\ K_i] = [30,\ 0.25,\ 1]$ based on the manual tuning result obtained previously. The motor side reference, θ_{md}, was selected to be a smoothed step input, i.e.

$$\theta_{md} = \frac{\omega_m^2}{s^2 + 2\omega_m s + \omega_m^2}\theta_{mo} \tag{3.21}$$

The magnitude of the reference input, θ_{mo}, was selected to be 1 radian, and the bandwidth of the smoothening filter was selected to be $\omega_m = 25\ rad/sec$. The parameters in the RTNLP method were chosen as follows: the cost function spans from $k_o = 1$ to $k_f = 1000$, the perturbation $\varepsilon = [0.01\ \sin(2\pi 0.427 i N_f t_s),\ 0.001\ \sin(2\pi 0.335 i N_f t_s),\ 0.01\ \sin(2\pi 0.371 i N_f t_s)]^T$, and the step size for updating the parameter $\gamma = [0.0003, 1.2 \times 10^{-7}, 5 \times 10^{-8}]^T$. Figure 3.6 shows the resulting cost

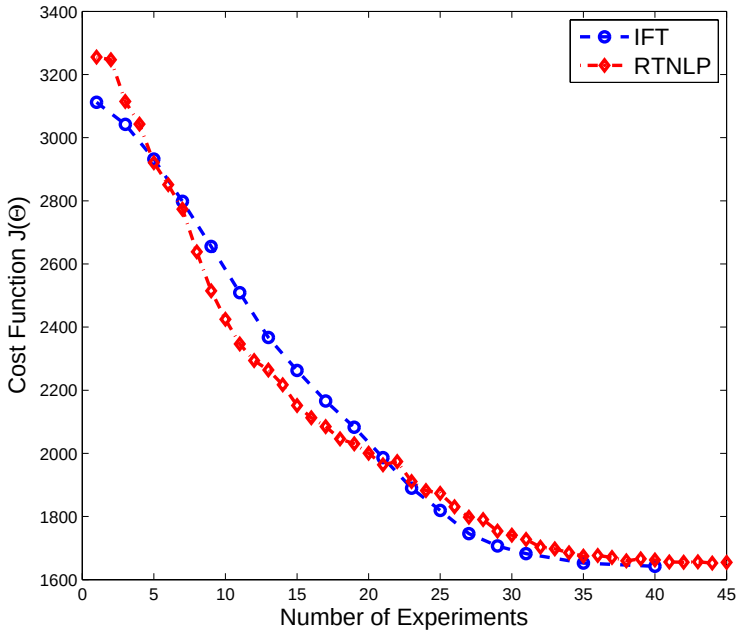


Figure 3.6: Minimization of the cost function in feedback controller tuning experiment

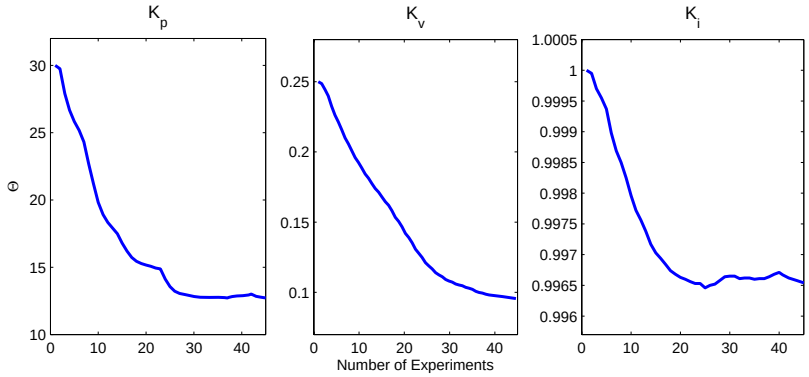


Figure 3.7: Tuning of PID controller parameters using RTNLP in experiment

functions in feedback controller tuning process. The circle-dashed line and the diamond-dashed line represent the evolution of the cost function for the IFT and the RTNLP method, respectively. It can be seen from the figure that the cost function converges after about 35 experiments for both methods. It is known that in real situations, every experiment is different from each other. Therefore, the initial values of the two cost functions are not identical in the figure. Notice that the RTNLP method requires one experiment for each iteration, while the IFT method, requires two experiments for each iteration: one normal experiment and one gradient experiment [67].

The step size parameters, γ, in the tuning algorithms were selected such that both algorithms have approximately the same convergent rate. The convergent rate for both methods can be further improved by increasing γ. In the experiments, N_f=1000. The cost function is penalizing during the whole time in one iteration and each experiment takes 1 second. Therefore, it requires less than one minute for the controller parameters to converge. Figure 3.7 shows the the convergence of feedback controller parameters using the RTNLP method. The proportional gain, K_p, was greatly reduced in order to suppress vibration on the load side of the indirect drive train setup. The PID parameters determined by the two tuning methods are presented in Table 3.1.

Table 3.1: Feedback controller parameters after tuning

Tuning Method	K_p	K_v	K_i
IFT	8.312	0.1019	1.174
RTNLP	12.48	0.098	0.9965

The performance of the feedback controller parameters before and after tuning are shown in Figure 3.8. Figure 3.8(a) shows the load side acceleration responses and Figure 3.8(b) shows the motor side position responses. In Figure 3.8(a), the solid and the dash dotted line show the responses using the IFT and RTNLP method after 40 experiments, respectively. The dotted line shows the response of the initial controller with the controller gains obtained previously from manual tuning (MT). It can be seen from the figure that for both tuning methods, the load side vibration settles much faster while the magnitude of the acceleration error is greatly reduced. In Figure 3.8(b), the solid and the dotted lines show the motor side position reference and the response of the initial controller, respectively. The dashed and the dash dotted lines show the responses of the IFT and RTNLP tuning method, respectively. Notice that the overshoot and accuracy of the motor side position are penalized for reducing the load side vibration. From Figure 3.8, the performance of the controller parameters tuned by the RTNLP method is roughly equivalent to that by the IFT method, which implies that both methods sought the same local minimum of the

same cost function.

3.4.3 Feedforward Controller Tuning

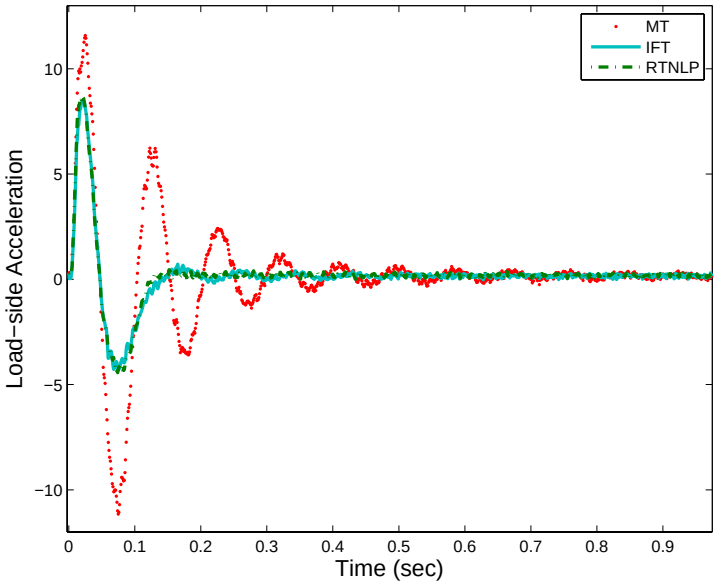
The feedforward controllers, C_{ff1} and C_{ff2}, are tuned after the feedback controller, C, has been tuned. In the tuning process of feedforward controller, the feedback controller parameters are fixed with the values obtained in the previous section. The feedforward controllers are tuned to further enhance the load-side position tracking performance. The proposed tuning method and the IFT method were used to tune the six feedforward gains, [K_{f1}, K_{f2}, K_{f3}, K_{f4}, K_{f5}, K_{f6}] in (3.3). The initial values of the controller parameters were set to be [0,0.01, 0, 0.08, 0, 0] based on (3.3). The parameters in the RTNLP method were chosen as follows: the cost function spans from $k_o = 1$ to $k_f = 1000$, the perturbation $\varepsilon = [1 \times 10^{-7} \sin(2\pi 0.401 i N_f t_s)$, $0.001 \sin(2\pi 0.335 i N_f t_s)$, $0.005 \sin(2\pi 0.298 i N_f t_s)$, $0.001 \sin(2\pi 0.371 i N_f t_s)$, 1×10^{-7} $\sin(2\pi 0.445 i N_f t_s)$, $1 \times 10^{-7} \sin(2\pi 0.247 i N_f t_s)]^T$, and the step size for updating the parameter $\gamma = [1 \times 10^{-14}, 9 \times 10^{-8}, 9 \times 10^{-7}, 9 \times 10^{-8}, 1 \times 10^{-14}, 1 \times 10^{-14}]^T$.

Trajectory generation

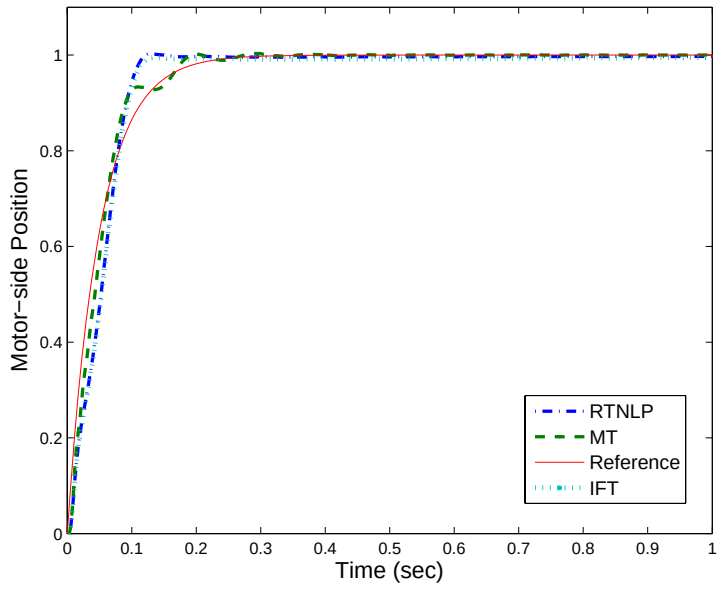
The reference trajectory is generated base on the idea of 4^{th} order time optimal trajectory design proposed in [42]. The planning algorithm is based on the construction of a derivative of jerk profile that can be integrated four times to obtain fourth order position trajectory. Assuming that a trajectory is planned for a point to point move over a distance, a symmetric trajectory is completely determined by four time intervals: the constant derivative of jerk interval $t_{\bar{d}}$, the constant jerk interval $t_{\bar{j}}$, the constant acceleration interval $t_{\bar{a}}$ and the constant velocity interval $t_{\bar{v}}$. The resulting profiles are given in Figure 3.9. Including the starting time of the trajectory at t_0, there are sixteen time instances at which the derivative of jerk changes:

$$\begin{aligned} t_{\bar{d}} &= t_1 - t_0 = t_3 - t_2 = t_5 - t_4 = t_7 - t_6 \\ &= t_9 - t_8 = t_{11} - t_{10} = t_{13} - t_{12} = t_{15} - t_{14} \qquad (3.22) \\ t_{\bar{j}} &= t_2 - t_1 = t_6 - t_5 = t_{10} - t_9 = t_{14} - t_{13} \qquad (3.23) \\ t_{\bar{a}} &= t_4 - t_3 = t_{12} - t_{11} \qquad (3.24) \\ t_{\bar{v}} &= t_8 - t_7 \qquad (3.25) \end{aligned}$$

By defining the level of the derivative of jerk, $\bar{d}$, and the four time intervals, $t_{\bar{d}}$, $t_{\bar{j}}$, $t_{\bar{a}}$, and $t_{\bar{v}}$, a trajectory is designed such that the fourth derivative of the load side position reference is guaranteed to exist. Figure 3.10 shows the desired load-side trajectory for feedforward controller tuning. It can be seen from the figure that in each iteration, the payload moves from the initial position to the desired position and back to the initial position in 2 seconds.



(a) $\ddot{\theta}_\ell(rad/sec^2)$



(b) $\theta_m(rad)$

Figure 3.8: Feedback controller tuning responses before and after

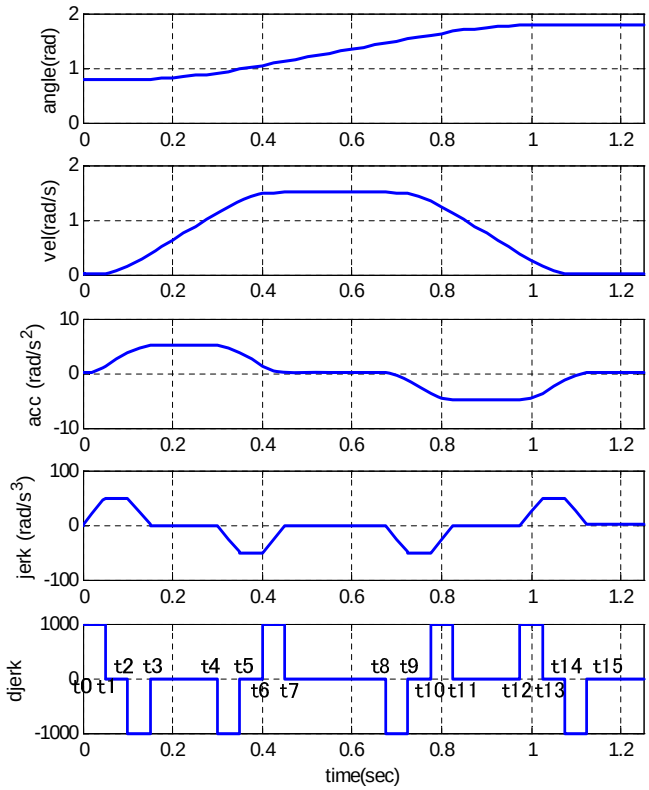


Figure 3.9: Fourth order trajectory profiles

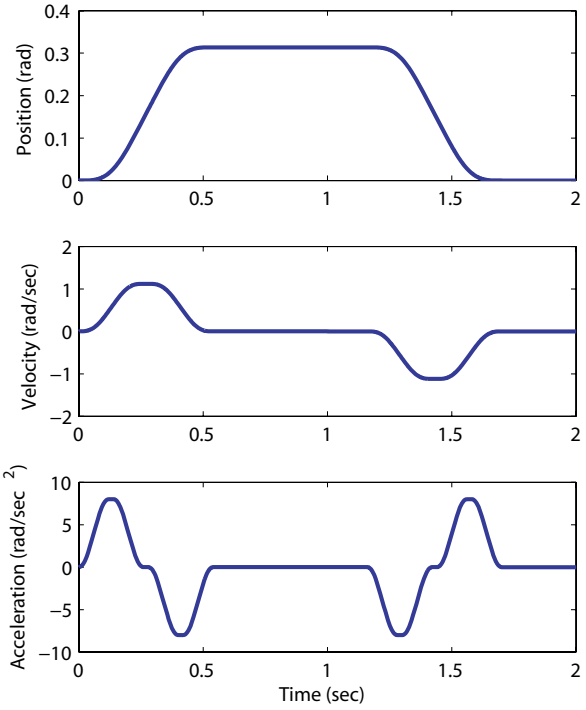


Figure 3.10: Load side reference trajectory

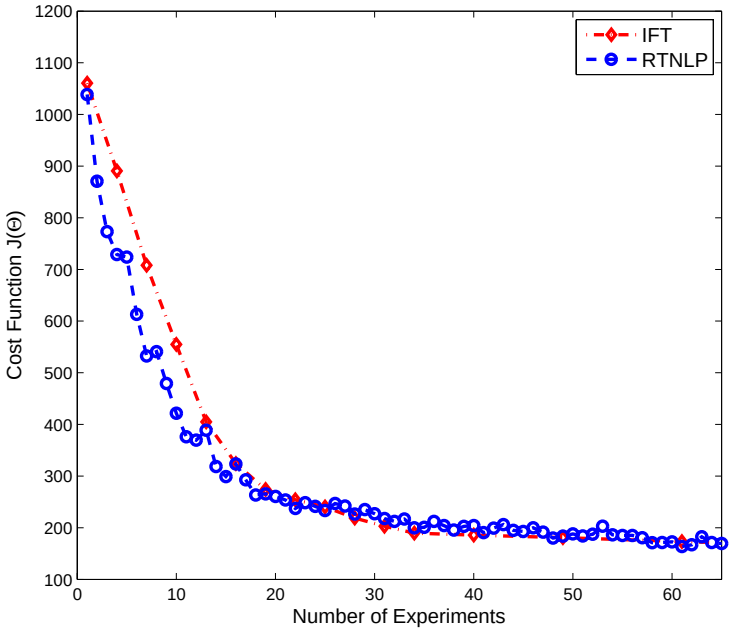


Figure 3.11: Minimization of the cost function in experiment (Feedforward controller tuning)

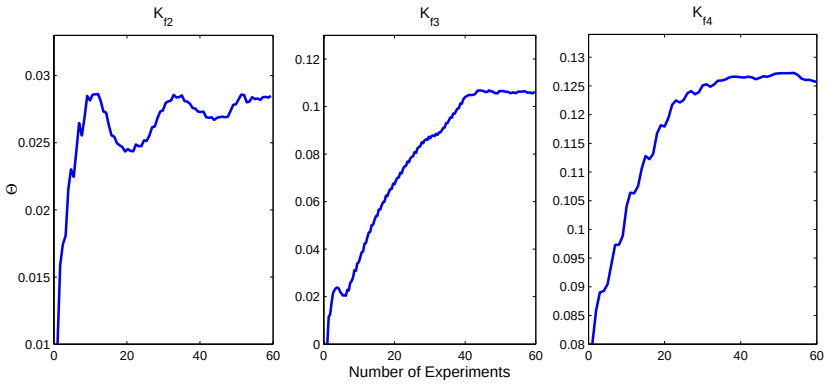


Figure 3.12: Tuning of feedforward controller parameters using RTNLP in experiment

Table 3.2: Feedforward controller parameters after tuning

Tuning Method	K_{f1}	K_{f2}	K_{f3}
IFT	2.17×10^{-5}	0.0284	0.1121
RTNLP	5×10^{-5}	0.0273	0.1064
Tuning Method	K_{f4}	K_{f5}	K_{f6}
IFT	0.1115	4.2×10^{-9}	1×10^{-8}
RTNLP	0.1257	2×10^{-5}	3×10^{-5}

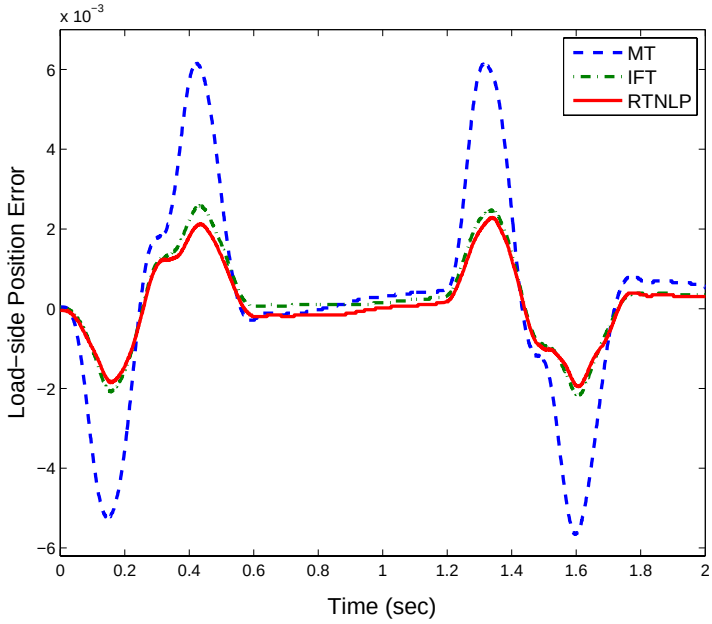
Results

Figure 3.11 shows the resulting cost functions for the IFT and the proposed method during the tuning process. It takes approximately 30 experiments for the cost functions to converge. Notice that in feedforward controller tuning process, the IFT method requires three experiments in each iteration: one normal experiment and two gradient experiments. The parameters for the IFT algorithm were selected such that both algorithms have approximately the same convergent rate. Figure 3.12 shows the convergence of the three feedforward controller parameters, K_{f2}, K_{f3}, and K_{f4}, using the proposed method. Only these three parameters are shown in the figure because they are the critical parameters that affect performance. The feedforward controller parameters determined by the two tuning methods are presented in Table 3.2.

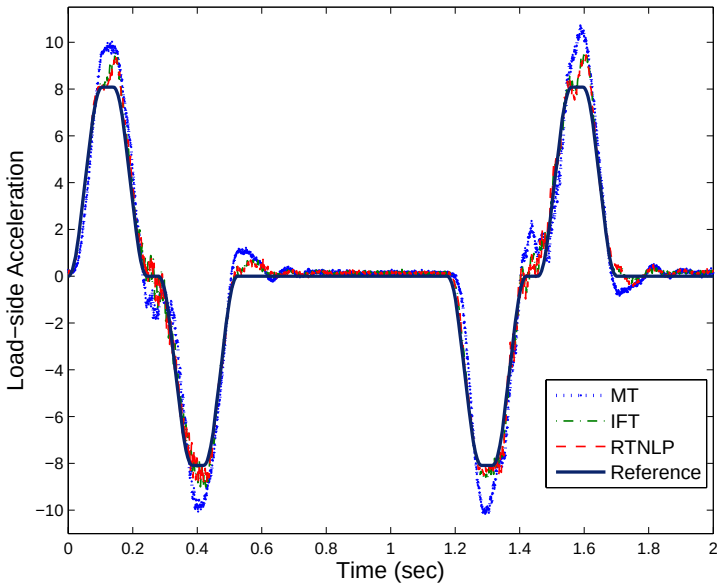
The tuning results are shown in Figure 3.13. Figure 3.13(a) shows the load side position tracking error responses while Figure 3.13(b) shows the load side acceleration responses. In Figure 3.13(a), the solid and the dash-dotted lines represent the tracking errors of the feedforward controller tuned by the proposed and the IFT method after 30 iterations, respectively. The dashed line shows the performance of the initial feedforward controller. Although the steady state error could not be completely eliminated because of the friction effect, the tracking performance has been significantly improved after tuning. In Figure 3.13(b), the solid and the dotted line shows the load-side acceleration reference and responses of the initial controller, respectively. The dashed and the dash dotted lines represent the acceleration response after tuning for the proposed and the IFT method respectively. It can be observed that the tuning of the controller reduces the acceleration tracking error as expected.

3.4.4 Selection of Parameters for the RTNLP method

Implementation of the proposed tuning method requires the choice of several parameters, namely, the perturbation amplitudes, a_l, adaptation gains, γ_i, perturbation



(a) $e_\ell(rad)$



(b) $\ddot{\theta}_\ell(rad/sec^2)$

Figure 3.13: Feedforward controller tuning responses before and after

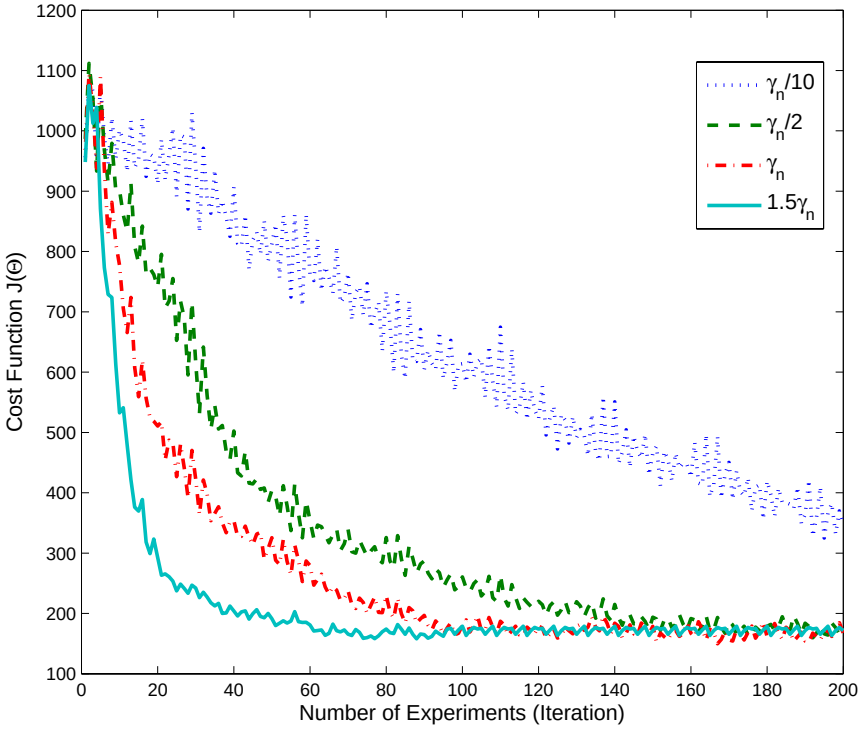


Figure 3.14: Evolution of the cost function during tuning of feedforward controller parameters using RTNLP with various values of γ ($\gamma_n = diag([5 \times 10^{-15},\ 5 \times 10^{-8},\ 5 \times 10^{-7},\ 5 \times 10^{-8},\ 5 \times 10^{-15},\ 5 \times 10^{-15}])$)

frequencies, ω_l, and the bandwidth of the low pass filter. The basic principle of how to choose these parameters was explained in Section 3.3. Through the experiments, it is discovered that the minimized cost value is fairly insensitive to the algorithm parameters. To investigate the sensitivity, the controller parameters of the indirect drive train system were tuned with various adaptation gains, γ_i. Figure 3.14 shows the evolution of the cost function during the feedforward controller tuning process. The tuning parameters other than γ are chosen to be those described in section 3.4.3. It can be seen from the figure that in each case, the cost function converges to a similar value with slower convergent rate for reduced gains. Table 3.3 shows that the proposed tuning algorithm yielded similar controller parameters even when γ was reduced by an order of 10.

Table 3.3: Feedback controller parameters for different adaptation speed γ after tuning ($\gamma_n = diag([5\times10^{-15},\ 5\times10^{-8},\ 5\times10^{-7},\ 5\times10^{-8},\ 5\times10^{-15},\ 5\times10^{-15}])$)

Parameters	K_{f1}	K_{f2}	K_{f3}
γ_n	5×10^{-5}	0.0263	0.1064
$1.2\gamma_n$	7×10^{-5}	0.0221	0.1142
$\gamma_n/2$	2×10^{-5}	0.0255	0.0961
$\gamma_n/10$	3×10^{-5}	0.0236	0.0922
Parameters	K_{f4}	K_{f5}	K_{f6}
γ_n	0.1257	2×10^{-5}	3×10^{-5}
$1.2\gamma_n$	0.1203	5×10^{-5}	3×10^{-5}
$\gamma_n/2$	0.1217	1×10^{-5}	5×10^{-5}
$\gamma_n/10$	0.1201	3×10^{-5}	3×10^{-5}

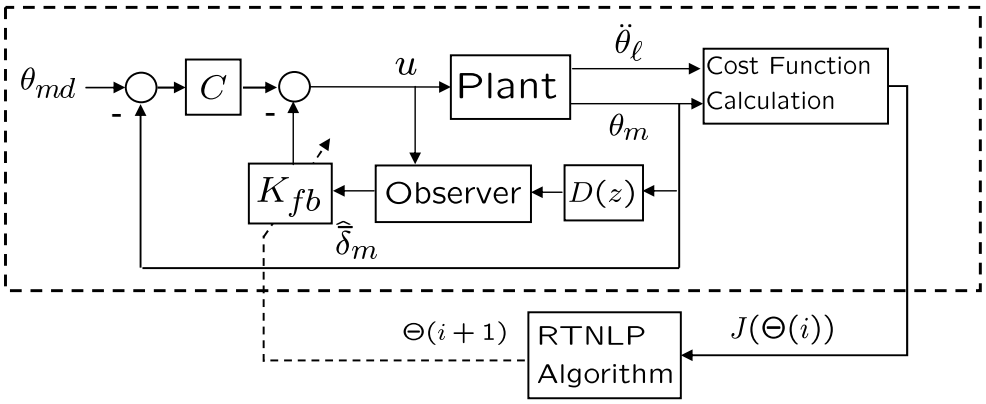


Figure 3.15: The state feedback controller tuning scheme

3.5 State Feedback Controller Tuning

3.5.1 Observer and State feedback Controller Structure

The proposed controller tuning scheme can also be used to tune the state feedback controller. The controller structure is shown in Figure 3.15 where C is the modified PID controller as discussed in Section 3.2 and K_{fb} is an inner torsion feedback controller with the feedback gain K_{fb} is $K_{fb} = [K_{fb1}\ K_{fb2}] \in \mathbf{R}^{1\times2}$. The inner torsion is estimated by an observer where the output of the observer is the estimate of the torsion, $\delta_m = \theta_m/N - \theta_\ell$, and the derivative of the torsion, $\dot{\delta}_m$.

Equation (2.4) shows that the dynamics of δ_m and $\dot{\delta}_m$ are not affected by the motor position θ_m. Therefore, if the motor velocity measurement is available, an ob-

server can be constructed based on the state-space representation (3.26) to estimate the torsion information, δ_m and $\dot{\delta}_m$.

$$\frac{d}{dt}\begin{bmatrix}\dot{\theta}_m\\ \delta_m\\ \dot{\delta}_m\end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{d_m}{J_m} & \frac{k_j}{J_m N^2} & \frac{d_j}{J_m N^2}\\ 0 & 0 & 1\\ \frac{d_\ell}{J_\ell}-\frac{d_m}{J_m} & -\frac{k_j}{J_m N^2}-\frac{k_j}{J_\ell} & -\frac{d_j}{J_m N^2}-\frac{d_j+d_\ell}{J_\ell}\end{bmatrix}}_{\mathbf{A}_o}\begin{bmatrix}\dot{\theta}_m\\ \delta_m\\ \dot{\delta}_m\end{bmatrix} + \underbrace{\begin{bmatrix}\frac{1}{J_m}\\ 0\\ \frac{1}{J_m}\end{bmatrix}}_{\mathbf{B}_o} u \tag{3.26}$$

$$y_o = \underbrace{\begin{bmatrix}1 & 0 & 0\end{bmatrix}}_{\mathbf{C}_o}\begin{bmatrix}\dot{\theta}_m\\ \delta_m\\ \dot{\delta}_m\end{bmatrix}$$

Equation (3.26) may be equivalently represented in the discrete-time domain as

$$\begin{bmatrix}\dot{\theta}_m(k+1)\\ \hline \delta_m(k+1)\end{bmatrix} = \underbrace{\left[\begin{array}{c|c}\mathbf{A}_{od11} & \mathbf{A}_{od12}\\ \hline \mathbf{A}_{od21} & \mathbf{A}_{od22}\end{array}\right]}_{\mathbf{A}_{od}}\begin{bmatrix}\dot{\theta}_m(k)\\ \hline \delta_m(k)\end{bmatrix} + \underbrace{\begin{bmatrix}\mathbf{B}_{od1}\\ \hline \mathbf{B}_{od2}\end{bmatrix}}_{\mathbf{B}_{od}} u(k) \tag{3.27}$$

$$y_o(k) = \mathbf{C}_{od}\begin{bmatrix}\dot{\theta}_m(k)\\ \hline \delta_m(k)\end{bmatrix} = \dot{\theta}_m(k)$$

where $\bar{\delta}_m = [\delta_m \ \dot{\delta}_m]^T$. It follows that the discrete-time reduced order observer can be constructed as

$$\begin{aligned}\hat{x}(k+1) &= (\mathbf{A}_{od22} - \mathbf{L}_d\mathbf{A}_{od12})\,\hat{x}(k) + (\mathbf{B}_{od2} - \mathbf{L}_d\mathbf{B}_{od1})\,u(k)\\ &\quad + [\mathbf{A}_{od21} - \mathbf{L}_d\mathbf{A}_{od11} + (\mathbf{A}_{od22} - \mathbf{L}_d\mathbf{A}_{od12})\,\mathbf{L}_d]\,y_o(k)\\ &:= \mathbf{A}_{ob}\hat{x} + \mathbf{B}_{uob}u(k) + \mathbf{B}_{yob}y_o(k)\\ \hat{\bar{\delta}}_m(k) &= \hat{x}(k) + \mathbf{L}_d y_o(k)\end{aligned} \tag{3.28}$$

where $\mathbf{L}_d$ is the observer gain.

3.5.2 Experimental Results

The proposed controller tuning method is used to tune the state feedback controller gains, $[K_{fb1},\ K_{fb2}]$, and the tuning scheme is shown in Figure 3.15. The experiment shown inside the dashed box is run iteratively as described in Section 3.3. The state feedback controller gains, $\Theta_{stfb}(i) = [K_{fb1}(i), K_{fb2}(i)]$, are adjusted in each iteration to minimize the performance index (3.6).

The feedforward controllers, C_{ff1} and C_{ff2}, are disabled during the state feedback controller tuning process. The feedback controller, C, is fixed during the tuning process of state feedback controller. The feedback controller gains, $[K_p,\ K_v,$

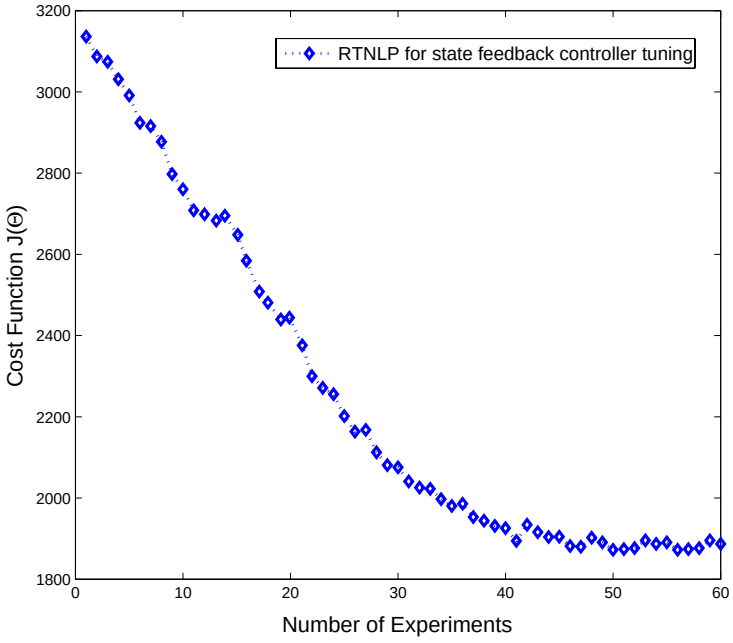


Figure 3.16: Minimization of the cost function in state feedback controller tuning experiment

K_i], were set to [30, 0.25, 1] and the initial values of the state feedback gains were set to zero. The reduced order observer (3.28) was designed by the pole placement method. The bandwidth of the observer was selected to be 20Hz. The motor side reference, θ_{md}, was selected to be 3.21.

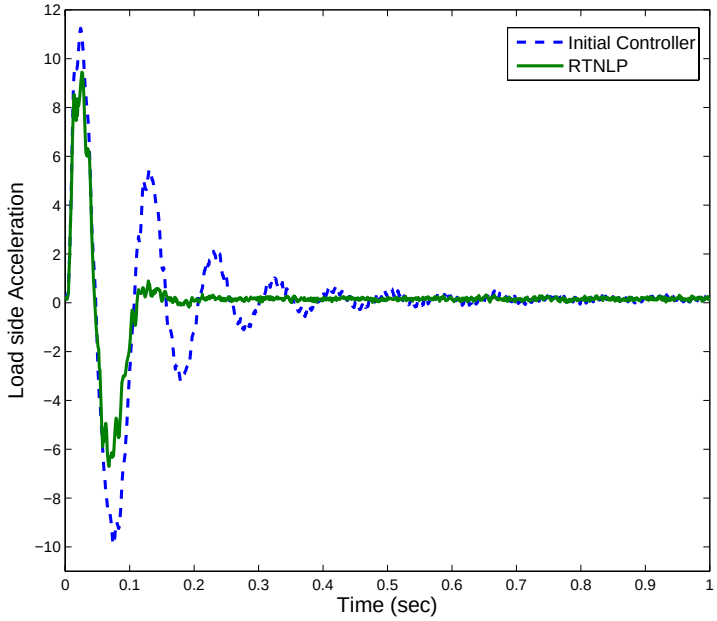
The parameters in the RTNLP method were chosen as follows: the cost function spans from $k_o = 1$ to $k_f = 1000$, the perturbation $\varepsilon = [0.005 \sin(2\pi 0.371 i N_f t_s)$, $0.0025 \sin(2\pi 0.445 i N_f t_s)]^T$, and the step size for updating the parameter $\gamma = [3 \times 10^{-6}, 3 \times 10^{-7}]^T$. Figure 3.16 shows the resulting cost function in state feedback controller tuning process. It can be seen from the figure that the cost function converges after about 45 experiments. Notice that the convergent rate for both methods can be further improved by increasing γ. The final controller gain, K_{fb}, was found to be $K_{fb} = [1.3428, -0.1295]$.

The performance of the state feedback controller parameters before and after tuning are shown in Figure 3.17. Figure 3.17(a) shows the load side acceleration responses and Figure 3.17(b) shows the motor side position responses. Comparing the acceleration response of the final controller to the one of the initial controller, it can be observed that the vibration has been greatly suppressed and the damping

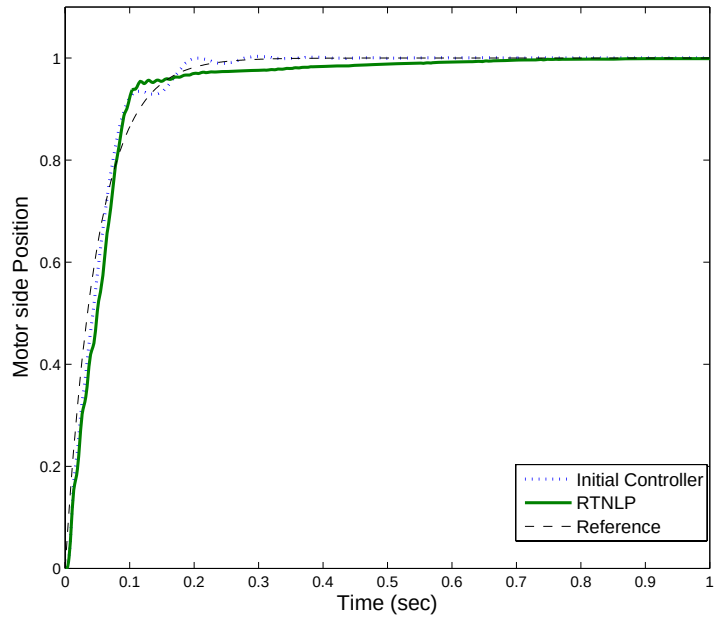
of the closed-loop system has been improved.

3.6 Summary

In this chapter, the controller parameters for a single-joint indirect train were optimized by real-time nonlinear programming which involved the estimation of the gradient of the cost function by perturbing the magnitude of the controller gains. Six feedforward, three feedback, and two state feedback controller gains were tuned. For the tuning of the state feedback controller gains, a reduced order observer was used to provide the torsion estimate. The load side performance was shown to be improved significantly after tuning. Moreover, unlike the IFT method, the RTNLP method used in this chapter did not require any additional experiments for the estimation of gradient.



(a) $\ddot{\theta}_\ell(rad/sec^2)$



(b) $\theta_m(rad)$

Figure 3.17: State feedback controller tuning responses before and after

Chapter 4

Suppression of Oscillations for Indirect drive Robots using Joint Sensing Information

4.1 Introduction

In indirect drive robots, harmonic drives are often used as the transmission mechanism for the purpose of speed reduction and torque amplification. They are designed such that several teeth are engaged at any given time making backlash virtually zero and hence are popular in applications requiring precision positioning such as the semiconductor industry.

A harmonic drive is composed of the components identified in Figure 4.1. The wave generator is an elliptically shaped steel core surrounded by a flexible race bearing. The circular spline is a rigid steel ring with teeth machined into the inner circumference. The flexspline is a thin-walled flexible cup having two fewer teeth on its outer rim than on the inner rim of the circular spline. Upon assembly, the wave-generator is inserted into the flexspline cup and the circular spline teeth then mesh with the flexspline teeth at the major axis of the ellipse defined by the wave generator.

The most common configuration for the harmonic drive is the speed reduction/torque magnification arrangement. This mode of operation usually consists of the wave generator as the input port, the flexspline as the output port, and the circular spline fixed to ground. In this configuration, the wave generator rotation corresponds to the motor angle input while the rotation of the flexspline in the opposite direction corresponds to the load side output.

Figure 4.2 shows the principle of motion of a harmonic drive in this configuration. In Figure 4.2(a), the flexspline is deflected by the wave generator into an elliptical shape causing the flexspline teeth to engage with those of the circular spline at

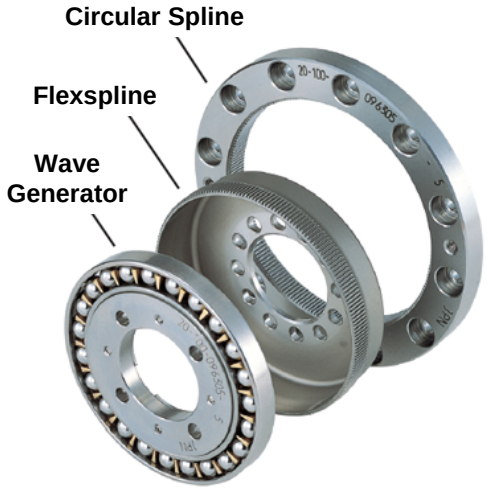


Figure 4.1: Harmonic drive gear components [26]

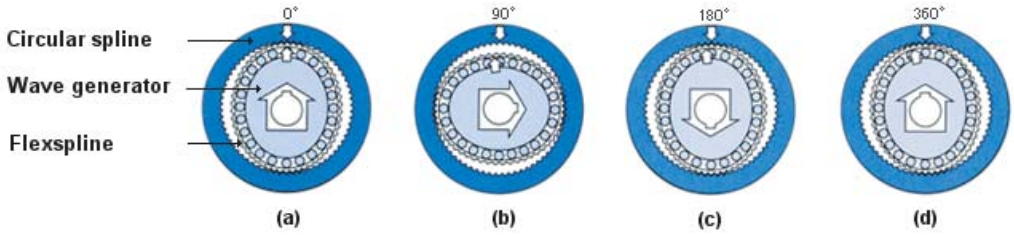


Figure 4.2: Principles of motion [26]

the major axis of the wave generator ellipse, with the teeth completely disengaged across the minor axis of the ellipse. When the wave generator is rotated clockwise with the circular spline fixed as shown in Figure 4.2(b), the flexspline is subjected to elastic deformation and its tooth engagement position moves by turns relative to the circular spline. When the wave generator rotates 180 degrees clockwise, the flexspline moves counterclockwise by one tooth relative to the circular spline as shown in 4.2(c). Figure 4.2(d) shows the situation when the wave generator rotates one revolution clockwise (360 degrees). The flexspline moves counterclockwise by two teeth relative to the circular spline because the flexspline has two fewer teeth than the circular spline. The special configuration of the harmonic drive makes them ideal for use in robotic applications which require high torque-to-weight ratio.

The concept of harmonic drives was conceived and developed during the mid-1950s [65]. Their industrial use for different applications such as robot manipula-

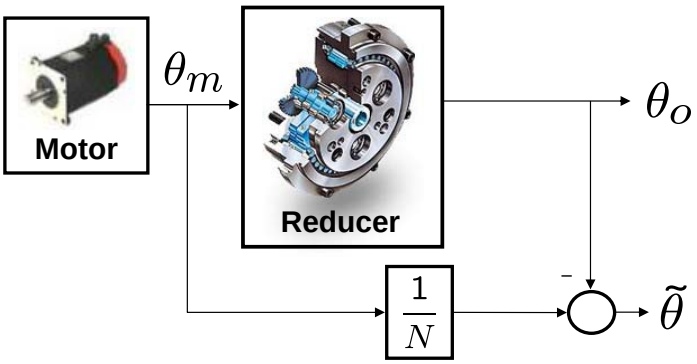


Figure 4.3: Definition of the transmission error

tors [29], force-feedback haptic devices [62] and steer by wire systems in the vehicle steering technology [3] has been growing since then. Research in the theoretical aspects of their transmission characteristics has been carried out in nonlinear transmission attributes including friction, transmission error, flexibility, and hysteresis [19, 50, 60].

Among different nonlinear characteristics mentioned above, the transmission error is of foremost concern for precision positioning applications. The transmission error of the gear introduces an inaccuracy into the kinematics chain and a positioning error due to the inaccuracy appears on the output of the gear. In many applications, the positioning error itself is not significant, but its influence on speed variation is critical. A transmission error causes a speed ripple on the gear output shaft, even at a constant rotation speed of the input shaft. These vibrations become dominant when the frequency of the error coincides with the resonant frequency of the control system. Speed fluctuations are largely amplified in such case and often exceed allowable level for vibrations[60, 69]. Thus reducing the vibration caused by the transmission error is important for precision tracking and performance.

The transmission error, $\tilde{\theta}$, is defined as the deviation between the expected output position and the actual output position of a harmonic drive. It is given by the following equation as shown in Figure 4.3

$$\tilde{\theta} = \frac{\theta_m}{N} - \theta_o \tag{4.1}$$

The experimental transmission error waveforms has been presented in the literature [60]. It shows a periodic nature with small amplitude. Due to mechanical imperfections such as gear assembly misalignments and dimensional inaccuracies of the gear itself, the output oscillations may vary with different drives and operating conditions. It has been shown in the literature, however, that the dominant component of the experimental transmission error waveforms is repeated every half

turn of the input shaft [19]; i.e. the ripple is periodic in nature and its fundamental frequency component corresponds to twice the rotation frequency of the input shaft. Hence, $\tilde{\theta}$ can be approximated with a simple sinusoid as follows

$$\tilde{\theta}_{dominant} = A_{te} \sin(\omega_o t + \phi) \tag{4.2}$$

where A_{te} is the amplitude of the transmission error which can be found in the harmonic drive catalogue. ϕ is the phase of the error and ω_o is the frequency of the error which is twice the rotation frequency of the motor shaft. Note that in (4.2) the higher frequency components in the error were neglected.

The velocity oscillation on the load side is highly undesirable for industrial robots. Therefore, the servo control system should have a control algorithm to suppress such oscillatory phenomena. In order to suppress the oscillations due to the transmission error, several control methods have been proposed in the literature[18, 20, 28]. Gandhi and Ghorbel [18] proposed nonlinear control algorithms for the compensation of the transmission error due to harmonic drives in set point regulation and trajectory tracking. Hirabayashi et al. [28] proposed a method where the controller senses speed ripples through a high resolution encoder and modifies the speed command to the driving motor. Godler et al. [20] applied the repetitive control for reducing speed ripples in a harmonic drive system. Among the different approaches, motor side information is widely used for feedback control. However, the velocity fluctuation due to the transmission error is most significant on the load side and the feedback controller cannot effectively reduce the vibrations on the load side by using only motor side information.

Adaptive feedforward cancellation (AFC) is an effective method for eliminating a periodic input/output disturbance, in which the disturbance is simply cancelled by adding the negative of its value at the input/output of the plant[54]. Since the value of the disturbance is generally unknown, it is adaptively determined by estimating the amplitudes of sine and cosine functions at disturbance frequencies using the output error signal. In this chapter, an adaptive disturbance cancellation scheme and its modifications are proposed and utilized to reduce the oscillations caused by the transmission error. Moreover, the load side acceleration feedback is used in the adaptation algorithms in order to minimize the load side vibrations effectively.

4.2 Dynamic Model of Single-Joint Indirect Drive Train with Transmission Error

Several harmonic drive models have been developed in the literature [18, 19]. In [19], a harmonic drive model considering transmission error and flexibility was developed using Lagrange formulation and in [18] the singular perturbed model of a

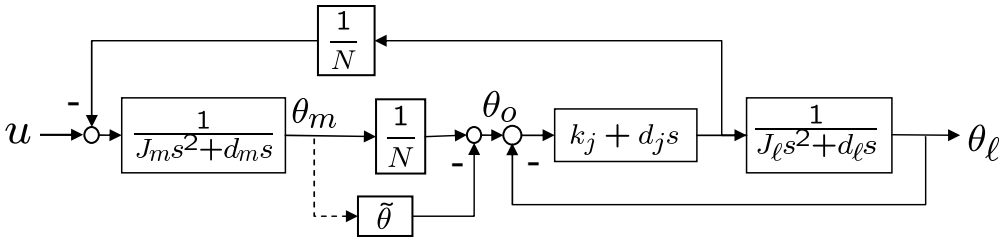


Figure 4.4: Block diagram of a single-joint indirect drive train with transmission error

harmonic drive considering transmission error and flexibility effect was derived. It has been shown in [66] that the model of the single-joint indirect drive train with the consideration of transmission error as shown in Figure 4.4 is the simplest yet accurate enough for representing the actual system. Note that as shown in the figure, the transmission error is included in the system as a position error at the output of the motor shaft. The dynamic model of the system shown in Figure 4.4 can be obtained as

$$\begin{aligned} J_\ell \ddot{\theta}_\ell + d_\ell \dot{\theta}_\ell &= k_j \left(\frac{\theta_m}{N} - \theta_\ell - \tilde{\theta} \right) + d_j \left(\frac{\dot{\theta}_m}{N} - \dot{\theta}_\ell - \dot{\tilde{\theta}} \right) \\ J_m \ddot{\theta}_m + d_m \dot{\theta}_m &= u - \frac{1}{N} \left[k_j \left(\frac{\theta_m}{N} - \theta_\ell - \tilde{\theta} \right) + d_j \left(\frac{\dot{\theta}_m}{N} - \dot{\theta}_\ell - \dot{\tilde{\theta}} \right) \right] \end{aligned} \tag{4.3}$$

By grouping the nonlinear terms due to the transmission error together, (4.3) can be viewed as the combination of two terms: the standard single-joint indirect drive train model in (2.3) and the nonlinear term that is due to the transmission error. It follows that the state-space representation of the system can be written as

$$\dot{x} = \mathbf{A}x + \mathbf{B}u + \mathbf{B}_d d \tag{4.4}$$

where

$$\mathbf{B}_d = [\ 0\ \tfrac{1}{J_m N}\ 0\ -\tfrac{1}{J_\ell}\ ]^T, \quad d = k_j \tilde{\theta} + d_j \dot{\tilde{\theta}}$$

$\mathbf{A}$, $\mathbf{B}$, x, are the same as those described in Section 2.3. Notice that with the formulation shown in (4.4), the transmission error is now contained in the disturbance input, d, and it affects the system through $\mathbf{B}_d$. Figure 4.5 shows the block diagram of the closed-loop system with the plant formulation shown in (4.4). r_m is the motor position reference and θ_m is the motor position. $\ddot{r}_\ell$ is the load side acceleration reference and $\ddot{\theta}_\ell$ is the load side acceleration. $P_\ell(s)$ and $P_m(s)$ are the transfer functions from the input u to the outputs $\ddot{\theta}_\ell$ and θ_m, respectively while

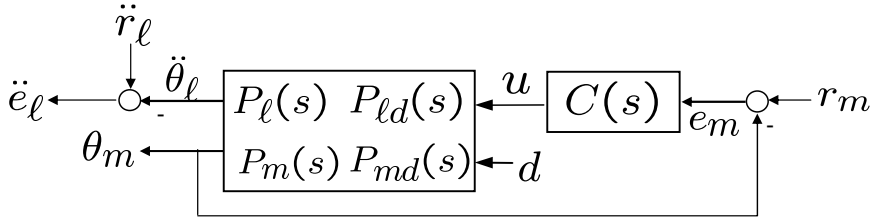


Figure 4.5: Block diagram of the closed-loop control system

$P_{\ell d}(s)$ and $P_{md}(s)$ are the transfer functions from the disturbance d to the outputs $\ddot{\theta}_\ell$ and θ_m, respectively. $\ddot{e}_\ell = \ddot{r}_\ell - \ddot{\theta}_\ell$ is the load side acceleration error. $C(s)$ is a modified PID controller given in Section 3.2.

Figure 4.6 shows the experimental results of the system in Figure 4.5 where the plant is replaced by the single-joint indirect drive setup described in Section 2.2. Figure 4.6(a) shows the time response of the load side acceleration error, $\ddot{e}_\ell$, when the motor reference r_m is designed such that the desired motor speed is maintained constant at 34 rad/sec. Figure 4.6(b) shows the load side acceleration error amplitude spectrum. Ideally, the load side acceleration should remain zero at the steady state but in real situation, a periodic error with the frequency at twice the motor rotation speed and the higher harmonics can be observed on the load side acceleration as shown in the figure. Notice that the first harmonic has the largest error amplitude.

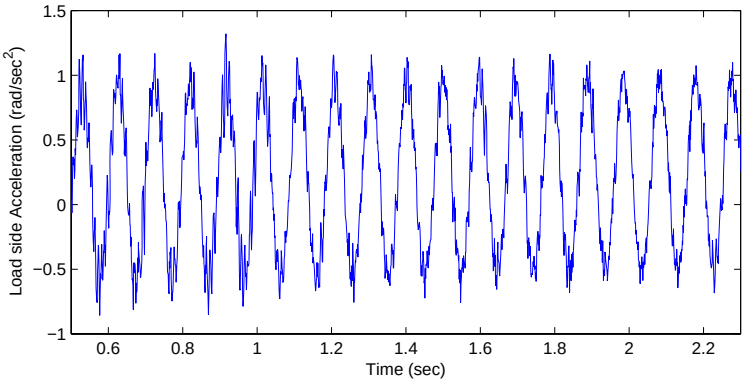
4.3 Transmission Error Compensation Algorithms

4.3.1 Adaptive Disturbance Cancellation Scheme

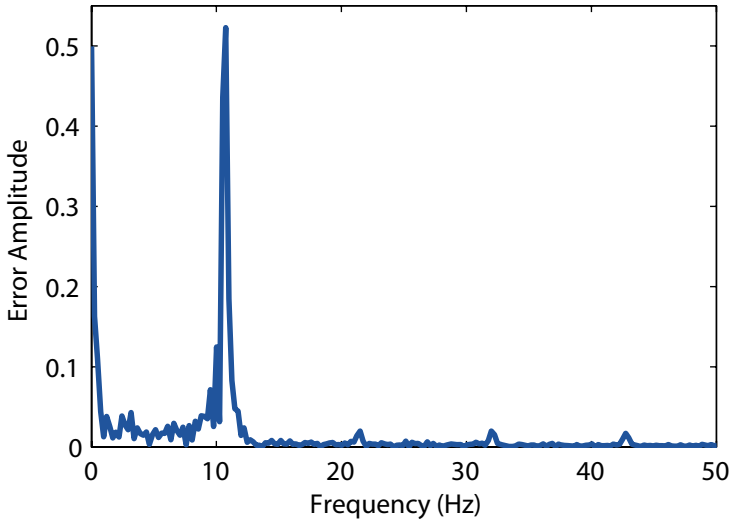
The general structure of the continuous-time adaptive compensation is shown in Figure 4.7. From (4.2) and (4.4), the time-varying unknown disturbance, $d(t)$, can be written as a sum of N_s sinusoids of known frequencies as follows

$$\begin{aligned} d(t) &= k_j\tilde{\theta} + d_j\dot{\tilde{\theta}} \\ &= \sum_{i=1}^{N_s}[k_j A_i \sin(\omega_{oi}t+\phi) + d_j A_i \omega_{oi}\cos(\omega_{oi}t+\phi)] \\ &= \sum_{i=1}^{N_s}[\alpha_i(t)\cos(\omega_i t) + \beta_i(t)\sin(\omega_i t)] \end{aligned} \tag{4.5}$$

where $\alpha_i(t)$ and $\beta_i(t)$ are the Fourier coefficients and are estimated in real-time. The subscripts i refers to the i-th frequency component of the disturbance. In the



(a) $\ddot{\theta}_\ell$



(b) Error Amplitude Spectrum

Figure 4.6: Load side angular acceleration response before compensation

compensation scheme, it is desired to remove the vibrations observed at the output of the system by forming a control input, $u_f(t)$, that exactly cancels the vibrations.

Notice that the compensation scheme shown in Figure 4.7 is different from the conventional AFC scheme where the disturbance, $d(t)$, is usually an additive output disturbance. Hence, in order to cancel the vibration caused by the input disturbance, the control input has the form

$$u_f(t) = \sum_{i=1}^{N_s} [\hat{a}_i(t)\cos(\omega_i t) + \hat{b}_i(t)\sin(\omega_i t)] \tag{4.6}$$

where the estimates of the coefficients must be adapted to the nominal values, i.e.

$$\begin{aligned} \hat{a}_i(t) &= a_i(t) \\ \hat{b}_i(t) &= b_i(t) \end{aligned}$$

Notice that $\mathbf{B} \neq \mathbf{B}_d$ in (4.4) implies that $a_i(t) \neq \alpha_i(t)$, and $b_i(t) \neq \beta_i(t)$. An adaptive algorithm to adjust the estimates $\hat{a}_i(t)$ and $\hat{b}_i(t)$ consists of the update laws [27]

$$\begin{aligned} \frac{d}{dt}\hat{a}_i(t) &= \gamma_i \ddot{e}_\ell(t)\cos(\omega_i t) \\ \frac{d}{dt}\hat{b}_i(t) &= \gamma_i \ddot{e}_\ell(t)\sin(\omega_i t) \end{aligned} \tag{4.7}$$

where the parameter γ_i is the constant adaptation gain, i.e. the step-size. The load side acceleration error, $\ddot{e}_\ell(t)$, is used to update the coefficients $\hat{a}_i(t)$ and $\hat{b}_i(t)$ correctly so that the compensation signal, $u_f(t)$, may suppress the oscillation shown on the load side effectively.

Implementation of (4.7) requires a discrete time representation of the equations. In this chapter, (4.7) is implemented with the backward Euler approximation

$$\begin{aligned} \hat{a}_i(k) &= \hat{a}_i(k-1) + T\gamma_i \ddot{e}_\ell(k)(\cos\omega_i T k) \\ \hat{b}_i(k) &= \hat{b}_i(k-1) + T\gamma_i \ddot{e}_\ell(k)(\sin\omega_i T k) \end{aligned} \tag{4.8}$$

where T is the sampling time.

Analysis of the single frequency cancellation case

It was shown in [5] that, for the adaptive compensation scheme shown in Figure 4.7, there is an equivalent linear time-invariant (LTI) representation of the transfer function from $\ddot{e}_\ell$ to u_f. The resulting continuous-time transfer function for multiple harmonics is

$$C_{fm}(s) = \sum_{i=1}^{N_s} \gamma_i \frac{s}{s^2 + \omega_i^2} \tag{4.9}$$

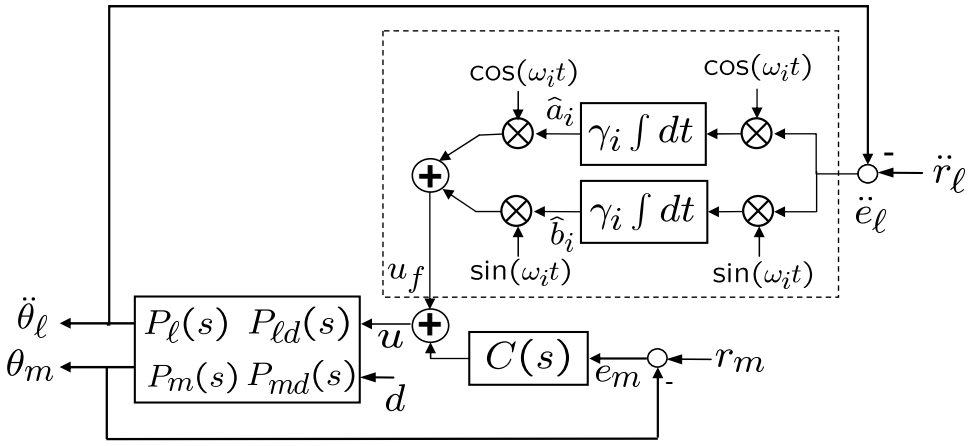


Figure 4.7: The proposed adaptive compensation scheme

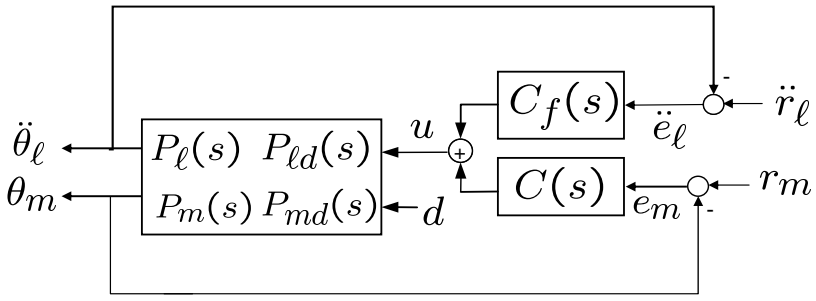


Figure 4.8: Block diagram of the proposed speed control system

Notice that when only canceling a single frequency disturbance, $C_{fm}(s)$ can be simplified as

$$C_f(s) = \gamma \frac{s}{s^2 + \omega^2} = \gamma C_{fo}(s) \tag{4.10}$$

where ω denotes the disturbance frequency, at which high disturbance rejection is required. Notice that $C_f(s)$ in (4.10) represents a peak filter with the center frequency at ω. The compensation scheme for the single frequency case is shown in Figure 4.8. With this structure, the analysis on the disturbance rejection and stability properties for the case of canceling a single frequency disturbance is shown below.

Without loss of generality, it is assumed that $C(s)$ is designed to assure the system's basic stability/performance requirements. A feedback loop with peak filter $C_f(s)$ using load side acceleration information is added to the original servo control system as shown in Figure 4.5. With this structure, the load side acceleration

feedback loop is connected to the motor feedback loop in a parallel form such that the filter $C_f(s)$ can be easily enabled or disabled whenever necessary. Moreover, the parallel connection considered here has a very useful interpretation in terms of shaping the original loop as discussed below.

Note that the sensitivity function from the disturbance input d to the load side acceleration error $\ddot{e}_\ell$ as shown in Figure 4.8 is given by:

$$\begin{aligned} S_\ell &= \frac{N_\ell}{1 + P_m C + P_\ell C_f} \\ &= \frac{N_\ell}{1 + P_m C} \frac{1 + P_m C}{1 + P_m C + P_\ell C_f} \\ &= S_{\ell o} S_f \end{aligned} \tag{4.11}$$

where

$$\begin{aligned} N_\ell &= P_{\ell d} + P_{\ell d} P_m C - P_\ell C P_{md} \\ S_{\ell o} &= \frac{N_\ell}{1 + P_m C} \\ S_f &= \frac{1}{1 + T_{\ell o} C_f} \\ T_{\ell o} &= \frac{P_\ell}{1 + P_m C} \end{aligned}$$

Note that $S_{\ell o}$ and $T_{\ell o}$ are the sensitivity function and complementary sensitivity function of the system shown in Figure 4.8, respectively. Equation (4.11) shows that the sensitivity function, S_ℓ, of the closed-loop system is the multiplication of two subsystem $S_{\ell o}$ and S_f , which implies that the controllers can be designed by a two-stage approach. In the first stage, the baseline controller $C(s)$ can be designed for basic closed-loop stability and disturbance rejection performance indicated by $S_{\ell o}$. In the second stage, the filter $C_f(s)$ can be designed based on the pseudo-plant $T_{\ell o}$ as shown in (4.11) such that S_f is shaped to a desired curve for rejecting disturbances in some frequency ranges. Noting that (4.10) and (4.11), the closed-loop system in Figure 4.8 will be stable if γ is chosen such that all the zeros of $Q(s)$ have negative real parts, where

$$Q(s) = 1 + \gamma T_{\ell o}(s) C_{fo}(s) \tag{4.12}$$

Note that C_{fo} was defined in (4.10).

Figure 4.9 shows the frequency response from the disturbance input, d, to the amplitude of the velocity error. The solid and dotted lines are the load side and motor side responses for the original control system, respectively. The dot and cross markers lines are the responses of the load side and motor side for the system with

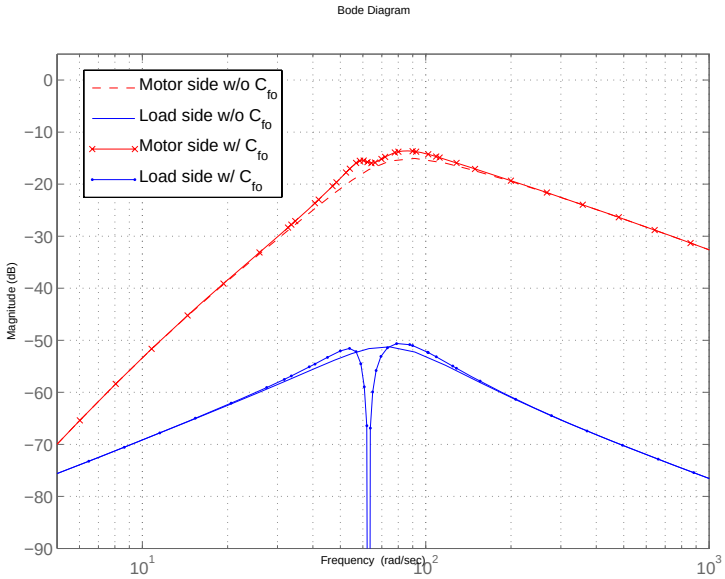


Figure 4.9: Frequency response from the disturbance input, d, to the velocity error before and after applying $C_f(s)$

filter, respectively. Notice that the filter, $C_{fo}(s)$, used in the figure has a center frequency, $\omega = 2\pi \cdot 11$ rad/sec in (4.10). It can be seen from the figure that the amplitude of the load side oscillation of velocity decreases significantly at the center frequency of the peak filter. The motor side velocity error is slightly enlarged, however, around the center frequency of the filter in order to compensate the effect of the transmission error on the load side.

4.3.2 Modified Schemes for Performance Enhancement

Compensation with Variable Step-Size Parameter

The purpose of the control system in Figure 4.7 is to reduce the amplitude of the load side vibration. From the LTI analysis, a larger γ_i would yield a smaller load side error magnitude at the steady state and a faster convergence rate of the update laws. As discussed in the previous section, however, the adaptation gain, γ_i, has an upper bound constrained by both the plant and the baseline controller $C(s)$ in order to guarantee stability. Moreover, similar to adding a peak filter in the feedback loop, having the adaptive cancellation scheme in the system drastically degrades the transient performance because of the pairs of complex poles near the

imaginary axis. The objective is to increase the noise attenuation while having a satisfactory transient response. To achieve this goal, a step-size adjustment method proposed in [41] is used here because of its simplicity and sensibility to changes in the error signal. The algorithm is controlled by the square of the error shown as follows

$$\gamma_i(k) = g_i\gamma_i(k-1) + \ell_i\ddot{e}_\ell^2(k-1) \tag{4.13}$$

The constant gain g_i is a forgetting factor with values between $0 < g_i < 1$ and $\ell_i > 0$ is the step-size parameter for the adaptation of γ_i. The algorithm adaptively changes the step-size, γ_i, of the update laws in (4.8) to achieve better vibration attenuation while having satisfactory transient performance.

Substituting the variable step-size, $\gamma_i(k)$, in the update equation for the filter coefficients (4.8), are obtained

$$\begin{aligned} \hat{a}_i(k) &= \hat{a}_i(k-1) + T\gamma_i(k)\ddot{e}_\ell(k)(\cos\omega_i Tk) \\ \hat{b}_i(k) &= \hat{b}_i(k-1) + T\gamma_i(k)\ddot{e}_\ell(k)(\sin\omega_i Tk) \end{aligned} \tag{4.14}$$

In this chapter, the initial step-size $\gamma_i(k)$ is set to zero and is bounded by the maximum value, γ_{max}, to ensure stability of the algorithm. The gains g_i and ℓ_i are chosen manually.

Compensation with the Consideration of Measurement Bias

In order to suppress the vibrations on the load side, the adaptation laws, (4.8) and (4.14), use the load side error information obtained from the accelerometers. The solid line in Figure 4.10 shows the experimental result of the acceleration response obtained from the accelerometer while doing motor side speed regulation. It can be seen from the figure that the measured signal is noisy and contains a bias. To deal with the bias problem, a Kalman filter accompanied by a low pass filter is proposed to estimate the magnitude of the bias.

The Kalman filter addresses the problem of estimating the state $x(k)$ of a discrete time process governed by the linear stochastic difference equation with a noisy measurement

$$\begin{aligned} x(k+1) &= \mathbf{A}x(k) + \mathbf{B}u_{KF}(k) + w(k) \\ y(k) &= \mathbf{C}x(k) + v(k) \end{aligned} \tag{4.15}$$

where the random variables $w(k)$ and $v(k)$ represent the process and measurement noise, respectively. In addition, they are assumed to be independent, white, zero mean, and normal random process with probability distributions where $p(w) \sim N(0, Q)$ and $p(v) \sim N(0, R)$.

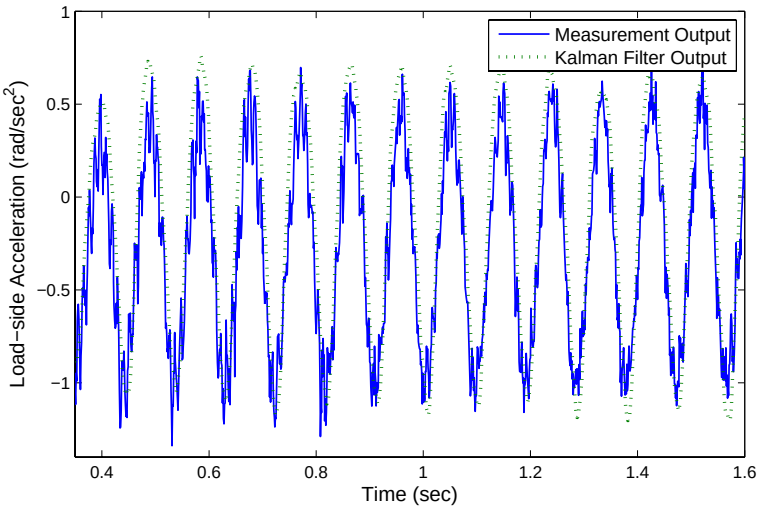


Figure 4.10: Comparison of responses from the measurement output and the Kalman filter output

In the Kalman filter setup, the following equations are utilized to calculate the state estimate $\hat{x}(k)$.
Time update:

$$\begin{aligned} \hat{x}(k+1|k) &= \mathbf{A}\hat{x}(k|k) + \mathbf{B}u_{KF}(k) \\ \mathbf{M}(k+1) &= \mathbf{A}\mathbf{P}(k)\mathbf{A}^T + Q \end{aligned} \tag{4.16}$$

Measurement update:

$$\begin{aligned} \mathbf{K}(k+1) &= \mathbf{M}(k+1)\mathbf{C}^T[\mathbf{C}\mathbf{M}(k+1)\mathbf{C}^T + R]^{-1} \\ \hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + \mathbf{K}(k+1)[y(k+1) - \mathbf{C}\hat{x}(k+1|k)] \\ \mathbf{P}(k+1) &= [\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}]\mathbf{M}(k+1) \end{aligned} \tag{4.17}$$

Notice that $\mathbf{K}(k)$ is the Kalman filter gain. $\mathbf{M}(k)$ and $\mathbf{P}(k)$ are the a priori and the a posteriori estimate error covariance, respectively.

Figure 4.10 shows the comparison of the measurement output and the Kalman filter output. It can be seen from the figure that Kalman filter is able to estimate the bias of the measured acceleration signal. Figure 4.11 shows the block diagram of the modified adaptive compensation system with the consideration of measurement bias. Kalman filter uses the information of measurement output and control input and then outputs the estimate, $\hat{\ddot{\theta}}_\ell$. After passing $\hat{\ddot{\theta}}_\ell$ through a low pass filter, the bias value, $\hat{\ddot{\theta}}_{\ell_{bias}}$, can be obtained. It is then subtracted from the real measurement, $\ddot{\theta}_\ell$, to generate the feedback signal to the adaptation algorithms in (4.8) and (4.14).

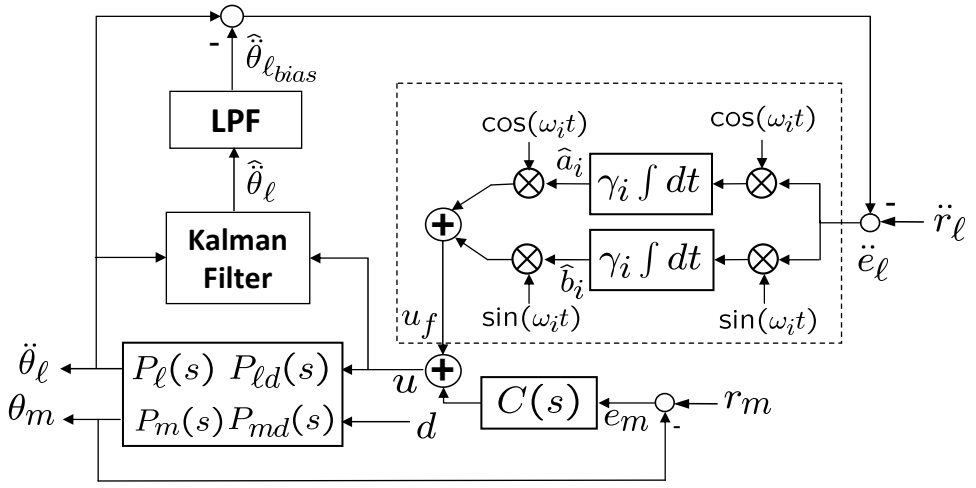


Figure 4.11: The proposed compensation scheme with the consideration of measurement bias

4.4 Experimental Study

4.4.1 Experimental Setup

The proposed methods are applied to a single-joint indirect drive train as described in Section 2.2.

For the experiments, three feedback controller gains, $[K_p, K_v, K_i]$ in (3.5) are set to [10, 0.3, 1]. Notice that for the controller structure used here, the control input, $u(k)$, contains a bias which corresponds to the control effort for compensating the friction effects. When implementing the Kalman filter, this control input bias must be removed to obtain an accurate acceleration bias estimate from the accelerometers. This can be easily done by subtracting the feedforward friction compensation signal from the actual control input, $u(k)$, i.e. $u_{KF}(k) = u(k) - f_m \text{sgn}(v_m)$ where f_m is the motor side friction coefficient and v_m is the motor velocity reference. f_m is set to 0.1004 in this chapter based on the previously identified result. The variance, Q and R, in the Kalman filter, (4.16) and (4.17), are set to 0 and 0.076, respectively. They are designed such that the Kalman filter can provide the accurate bias estimate. The cutoff frequency of the low pass filter is chosen to be below the disturbance frequency in order to extract the bias information from the acceleration measurement estimate.

The desired load side trajectory is shown in Figure 4.12. It is obtained by the 4^{th} order trajectory generation method explained in Section 3.4.3. The motor side reference is set to be the load side reference times the gear ratio. It can be seen from the figure that, the payload of the single-joint robot moves from the zero position to the desired position at a constant speed in 2.5 seconds and then back to the zero

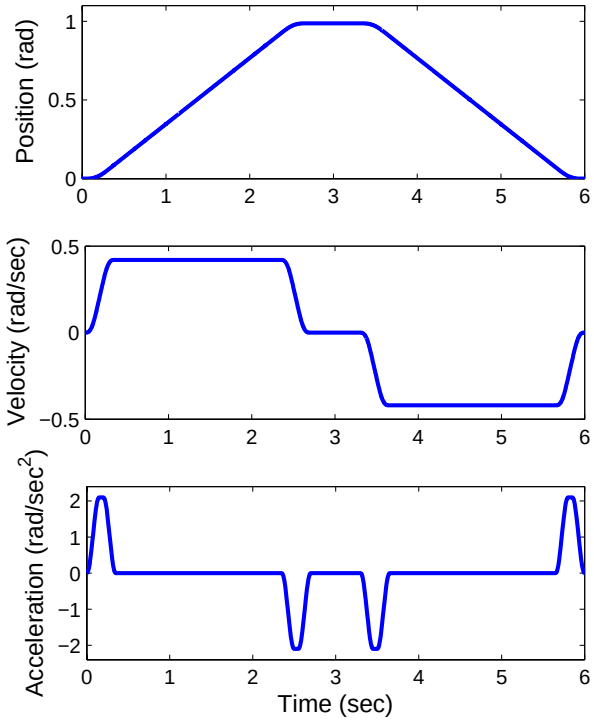


Figure 4.12: Load side reference trajectory

position.

Since the oscillation introduced by the harmonic drive is periodic with frequency twice the angular speed of the motor shaft, for simplicity, the sine and cosine functions in the compensation algorithms, (4.6), are modified to be $\sin(2\theta_m)$ and $\cos(2\theta_m)$, respectively. Moreover, the higher harmonics of the disturbance frequency are not considered in the implementation. Equations (4.6) and (4.8) become

$$\begin{aligned} u_f(k) &= \hat{a}(t)\cos(2\theta_m(k)) + \hat{b}(t)\sin(2\theta_m(k)) \\ \hat{a}(k) &= \hat{a}(k-1) + T\gamma\ddot{e}_\ell(k)(\cos 2\theta_m(k)) \\ \hat{b}(k) &= \hat{b}(k-1) + T\gamma\ddot{e}_\ell(k)(\sin 2\theta_m(k)) \end{aligned}$$

4.4.2 Experimental Results

Figure 4.13 shows the responses of the load side acceleration error of the original compensation algorithm, i.e. fixed step-size scheme, and the modified compensation

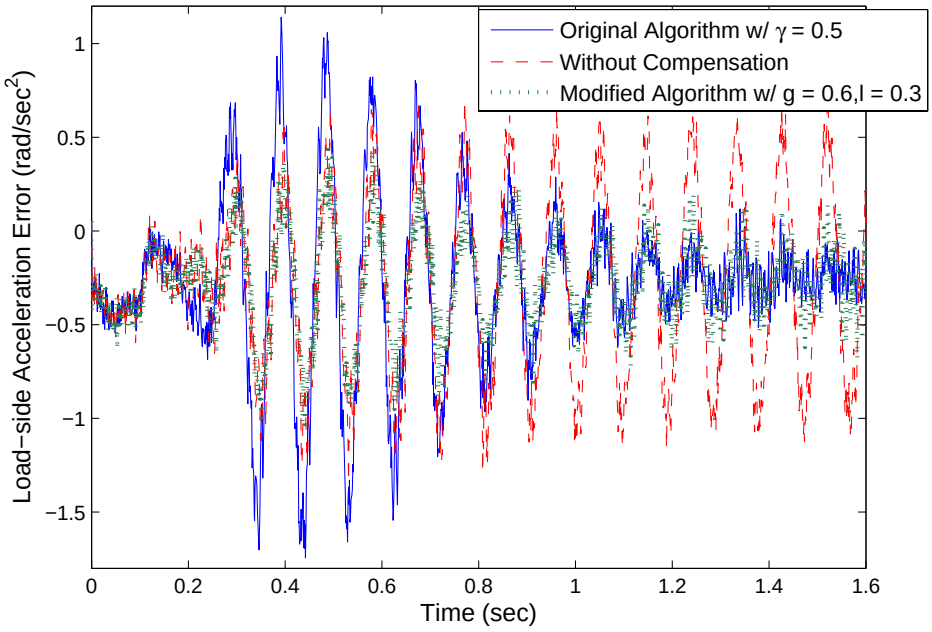


Figure 4.13: Load side acceleration error responses before and after

algorithm, i.e. variable step-size scheme. The dashed line represents the acceleration error without the compensation while the solid line, and the dotted line represents the error response of the system with the original and the modified compensation scheme, respectively. It can be seen from the figure that with the original algorithm, the acceleration error was amplified during transient response. The modified algorithm, on the other hand, adaptively changes the step-size of the update laws, (4.14), to attenuate the error. Notice that the step-size γ in the original algorithm and the parameters, g and ℓ, for adapting γ in modified algorithm were chosen so that the two compensation schemes achieve same level of error attenuation at steady state.

The load side acceleration error responses of the algorithms considering measurement bias are shown in Figure 4.14. The solid and dotted lines represent the error response of system with the original and with the modified compensation scheme, respectively. It is observed from the figure that removing the bias of the acceleration measurement by a Kalman filter, (4.16) and (4.17), further improves the convergence rate. Notice that now with the original compensation scheme, the error is not amplified during transient while still maintaining the same level of vibration reduction performance as the modified scheme at the steady state.

Figure 4.15 shows the adaptation transient of the Fourier coefficients, $\hat{a}(t)$ and

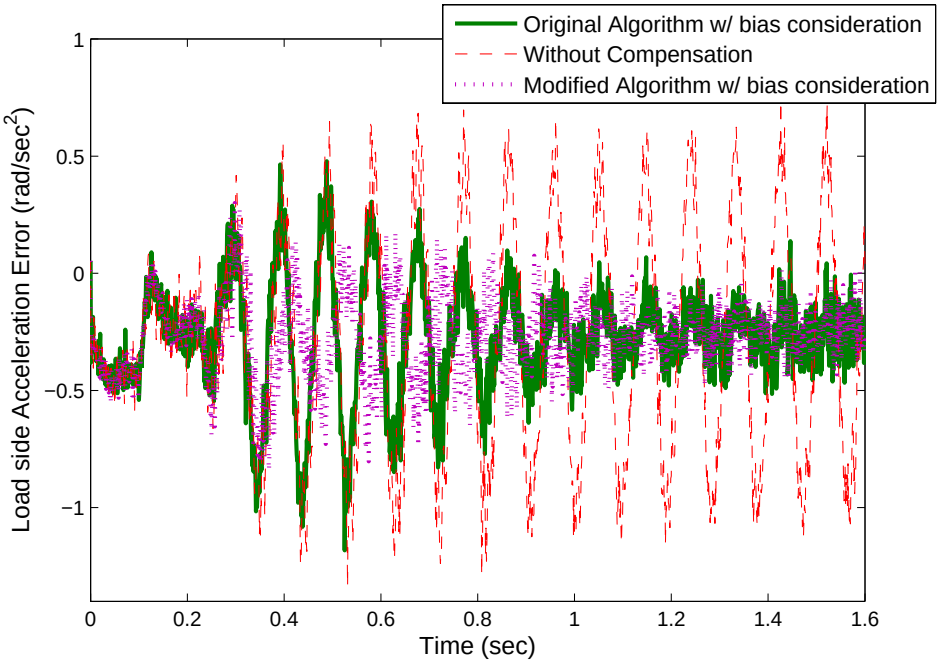


Figure 4.14: Load side acceleration error responses considering measurement bias before and after

$\hat{b}(t)$. It can be seen from the figure that the convergence rate of algorithms with the consideration of measurement bias is much faster than the ones without. Moreover, by removing the bias term in the measurement, the Fourier coefficients can be estimated more accurately to compensate for the input disturbance. Notice that, for the same trajectory, the coefficients $\hat{a}(t)$ and $\hat{b}(t)$ for each algorithm both converge to similar values but not exactly the same. Other than the bias and measurement noise discussed above, the phase differences while performing each experiment can also contribute to this phenomenon. Figure 4.16 shows the amplitude spectrum of the load side position error from the load side encoder on the single-joint indirect drive setup. It confirms the effectiveness of the proposed methods and the modified algorithms.

4.5 Summary

In this chapter, an adaptive cancellation algorithm was presented to suppress the oscillatory phenomena caused by the transmission error of harmonic drives. The acceleration information was used in the adaptation laws to effectively reduce the

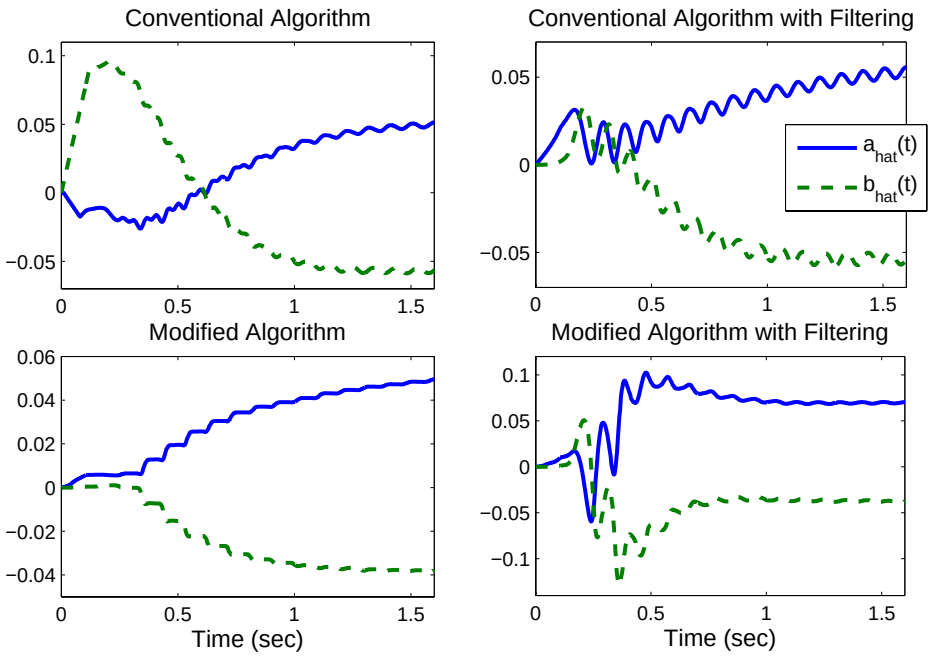


Figure 4.15: Estimation of the Fourier series coefficients $\hat{a}(t)$ and $\hat{b}(t)$

oscillations on the load side. Two modification schemes were proposed to deal with the deteriorate transient response and acceleration measurement bias. Experimental results showed the effectiveness of the proposed methods, and the load side oscillation due to the transmission error was reduced significantly.

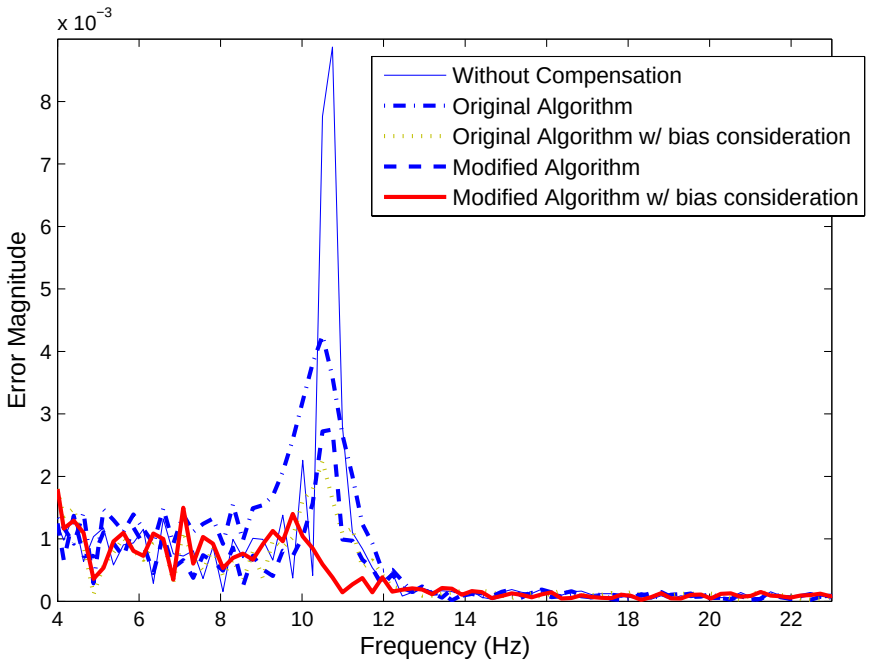


Figure 4.16: Amplitude spectrum of the load side position error

Chapter 5

Iterative Learning Control for Indirect Drive Trains

5.1 Introduction

In industrial applications, a robot often performs the same task repeatedly. When the robot follows a trajectory repeatedly using the same control every cycle, the trajectory tracking error will become repetitive from one trial to another. In such situation, the iterative learning control (ILC) method becomes promising and practical. ILC is a well-established method for control of repetitive processes. It was first introduced in 1970s and became popular in 1980s when Arimoto [1] proposed the learning schemes for the robotic applications. Some other examples of applying iterative learning control on robot manipulators can be found in [33, 43].

In the literature, iterative learning control is in general considered to be an approach for the purpose of trajectory tracking [4, 7, 23, 45, 68]. In [68], the author studies the learning control based on the load side position measurement with slow sampling rates (e.g. vision sensors). Two approaches are provided to deal with such learning control problems: one approach uses a multirate Kalman smoother to obtain the load side position estimate at the desired sampling rate by fusing the position measurement and additional load side acceleration measurements at fast sampling rates and another approach utilizes an interlacing technique to obtain the slow sampling rate position measurement, and updates the learning controller at the desired sampling rate using only the available slow sampling rate measurement. The effectiveness of both approaches on the load side position tracking performance are demonstrated by experiments. In this chapter, however, a different setting is proposed by designing the iterative learning controller for the purpose of disturbance rejection. Disturbance rejection aspects of iterative learning control have been covered earlier in [11, 12, 48], where disturbances such as initial state variations and measurement disturbances are addressed.

The iterative learning controller uses the error information of the past cycles to correct control action in the current cycle. Clearly, the error signal plays a substantial role in learning control. In industrial applications, based on availability, the error signals can be chosen as the load side (end-effector) tracking errors, or motor side tracking errors. Although these errors are related to each other, the performance of the learning controllers using different error signals may be different. To apply the iterative learning control to the drive trains, the conventional approach is to assume that the robot is rigid, such that all relevant signals can be calculated from the motor angles. In [49], the motor side measurement is used in the learning controller without considering the gear effects. In reality, however, there are always mechanical flexibilities in both joints and links. Moreover, the performance of the drive train system is often evaluated by the load side position. If the load side sensor information is available, it is desired to use the load side measurement in the learning controller to directly compensate for the effects of the gear nonlinearities and other disturbances on the load side tracking performance. Different iterative learning control ideas applied to flexible mechanical systems have been studied previously in [43, 64]. In these papers, it is assumed that the load side position can be measured. In stead of such an assumption, it is assumed in this chapter that only the load side acceleration can be measured in addition to the motor side position and an estimation scheme is proposed to provide the estimated load side information.

This chapter is organized as follows. Section 5.2 presents the description of the system in the matrix form. Section 5.3 provides the problem formulation. The design of the motor side and load side iterative learning controller is presented in Section 5.4 and 5.5, respectively. Section 5.6 discusses the proposed estimation scheme for the load side position. The effectiveness of the learning controllers are shown by the experimental results in Section 5.7. Finally, Section 5.8 presents the conclusions.

5.2 Matrix Description of the System

The controller structure of the single-joint indirect drive train system with an iterative learning controller is shown in Figure 5.1. The indirect drive train is represented by the nominal plant models, G_ℓ and G_m, obtained in Section 2.2. r_ℓ(t), $\theta_\ell(t)$, and, $\theta_m(t)$ are the reference load side trajectory, the load side position, and the motor side position, respectively. The subscript j denotes the iteration number. The feedback controller, C, is a modified PID controller explained in Section 3.2 and the feedforward controller, C_{ff1}, is the filter that translates the load side reference trajectory, $r_\ell(t)$, to the motor side reference trajectory, $r_m(t)$, explained in Section 3.2. The iterative learning controller is shown in the dashed box. It uses the error signal, $e_{j-1}(t)$, and the compensation signal, $u_{c,j-1}(t)$ from the previous iteration

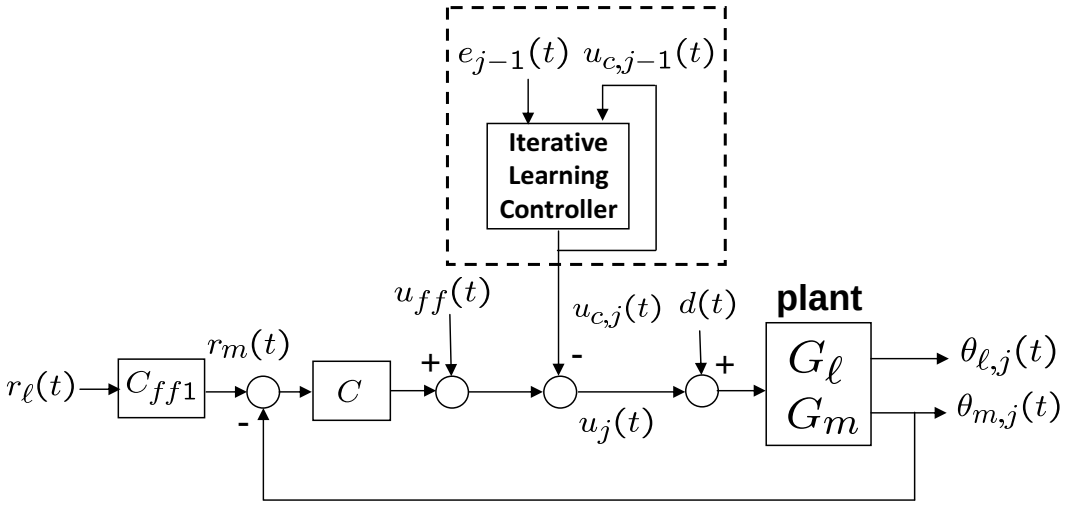


Figure 5.1: Controller structure of an indirect drive system with an iterative learning controller

and generates the output of the learning controller, $u_{c,j}(t)$, which is used as a complement input signal that directly alters the control signal to the plant [53]. $d(t)$ is the fictitious input disturbance that causes the vibrations in the system and $u_{ff}(t)$ is the feedforward torque which will be designed and explained in the following sections.

In Figure 5.1, the closed-loop transfer functions from the desired load side trajectory, $r_\ell(t)$, the learning compensation signal, $u_c(t)$, the feedforward torque, $u_{ff}(t)$, and the disturbance, $d(t)$, to the motor side position, $\theta_m(t)$, and the load side position, $\theta_\ell(t)$, can be obtained as

$$\begin{aligned}
\theta_{m,j}(t) &= \frac{G_m(s)C(s)C_{ff1}(s)}{1+G_m(s)C(s)}r_\ell(t) + \frac{G_m(s)}{1+G_m(s)C(s)}(d_j(t)+u_{ff}(t)-u_{c,j}(t)) \\
&= T_{mrc}(s)r_\ell(t) + T_{mc}(s)(d_j(t)+u_{ff}(t)-u_{c,j}(t)) \qquad (5.1) \\
\theta_{\ell,j}(t) &= \frac{G_\ell(s)C(s)C_{ff1}(s)}{1+G_m(s)C(s)}r_\ell(t) + \frac{G_\ell(s)}{1+G_m(s)C(s)}(d_j(t)+u_{ff}(t)-u_{c,j}(t)) \\
&= T_{\ell rc}(s)r_\ell(t) + T_{\ell c}(s)(d_j(t)+u_{ff}(t)-u_{c,j}(t)) \qquad (5.2)
\end{aligned}$$

Notice that in (5.2), s represents the operator d/dt in time domain. Equations (5.1) and (5.2) can be written in the discrete-time form by using the zero-order-hold (ZOH) method

$$\begin{aligned}
\theta_{m,j}(k) &= T_{mr}(q)r_\ell(k) + T_m(q)(d_j(k)+u_{ff}(k)-u_{c,j}(k)) \qquad (5.3) \\
\theta_{\ell,j}(k) &= T_{\ell r}(q)r_\ell(k) + T_\ell(q)(d_j(k)+u_{ff}(k)-u_{c,j}(k)) \qquad (5.4)
\end{aligned}$$

where k is the sampling point and q is the one step advance operator. Notice that in the learning control problem, we are only interested in the dynamics of the system defined over a finite interval of time. From (5.3) and (5.4), the relationship between the input and the resulting output time history can be formulated in the lifted form

$$\mathbf{Y}_{m,j} = \mathbf{T}_{mr}\mathbf{R}_\ell + \mathbf{T}_m(\mathbf{D}_j + \mathbf{U}_{ff,j} - \mathbf{U}_{c,j}) \tag{5.5}$$

$$\mathbf{Y}_{\ell,j} = \mathbf{T}_{\ell r}\mathbf{R}_\ell + \mathbf{T}_\ell(\mathbf{D}_j + \mathbf{U}_{ff,j} - \mathbf{U}_{c,j}) \tag{5.6}$$

where

$$\begin{aligned}
\mathbf{Y}_{m,j} &= [\theta_{m,j}(m), ..., \theta_{m,j}(m+N-1)]^T \\
\mathbf{Y}_{\ell,j} &= [\theta_{\ell,j}(m), ..., \theta_{\ell,j}(m+N-1)]^T \\
\mathbf{U}_{c,j} &= [u_{c,j}(0), ..., u_{c,j}(N-1)]^T \\
\mathbf{D}_j &= [d_j(0), ..., d_j(N-1)]^T \\
\mathbf{U}_{ff} &= [u_{ff,j}(0), ..., u_{ff,j}(N-1)]^T \\
\mathbf{R}_\ell &= [r_\ell(0), ..., r_\ell(N-1)]^T
\end{aligned}$$

m and N denote the system delay and the duration of the input signal, respectively. $\mathbf{T}_m$ is a lower triangular matrix formed by the impulse response coefficients of T_m. That is, if

$$T_m(q) = t_0 + t_1 q^{-1} + t_2 q^{-2} + \cdots \tag{5.7}$$

then the matrix form of $\mathbf{T}_m$ is constructed as

$$\mathbf{T}_m = \begin{pmatrix} t_0 & 0 & 0 & \cdots & 0 \\ t_1 & t_0 & 0 & \cdots & 0 \\ t_2 & t_1 & t_0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ t_{N-1} & \cdots & t_2 & t_1 & t_0 \end{pmatrix}, \tag{5.8}$$

$\mathbf{T}_{mr}$, $\mathbf{T}_\ell$, and $\mathbf{T}_{\ell r}$ can be defined analogously. In (5.8), the diagonal terms of the matrix is a constant. Moreover, it is assumed that the initial conditions are zero, i.e., $\theta_j(k) = 0$ for $k < 0$, and the system delay, m, is zero. Notice that if the system is time varying in the sense that the dynamics change during one iteration, it is straightforward to let the coefficients in (5.8) vary along the diagonals. It is, however, assumed here that the same $\mathbf{T}_m$, $\mathbf{T}_{mr}$, $\mathbf{T}_\ell$, and, $\mathbf{T}_{\ell r}$ are valid in each iteration.

5.3 An Optimization-based Approach to ILC

The optimization approach to iterative learning control has been developed for many years and the previous contributions can be found in [17, 21, 35]. The idea

of the optimization-based ILC is to determine the input, $\mathbf{U}_{c,j+1}$, for the system described in (5.5) and (5.6) such that the error, $\mathbf{E}_{j+1}$, becomes as small as possible by minimizing the criterion

$$\mathbf{J} = \mathbf{E}_{j+1}^T \mathbf{W}_e \mathbf{E}_{j+1} + \mathbf{U}_{c,j+1}^T \mathbf{W}_u \mathbf{U}_{c,j+1} \tag{5.9}$$

where $\mathbf{W}_e$ and $\mathbf{W}_u$ are the weighting matrices which determine the trade off between the performance and the input energy. $\mathbf{U}_{c,j+1}$ is the compensation signal generated by the learning controller. The error signal, $\mathbf{E}_{j+1}$, can be chosen as either the motor side or the load side error signal, if available.

The conventional optimization-based design of the iterative learning controller [49] is to use the motor side position error, $\mathbf{E}_{m,j} = \mathbf{R}_m - \mathbf{Y}_{m,j}$, as the error signal, $\mathbf{E}_j$, in (5.9). It is because the motor position measurement, $\mathbf{Y}_{m,j}$, can be easily obtained by the motor encoder and is often already used in the design of the feedback controller, C. In this approach, it is hoped to indirectly reduce the load side position error by improving the motor side performance. On the other hand, if the load side position measurement, $\mathbf{Y}_{\ell,j}$, is available, the load side tracking error, $\mathbf{E}_{\ell,j} = \mathbf{R}_\ell - \mathbf{Y}_{\ell,j}$, can be used as the error signal, $\mathbf{E}_j$, in (5.9). Based on the choice of the error signal, two different iterative learning controllers can be obtained. In this chapter, the learning controller is called as the motor side or the load side iterative learning controller, if the motor side or the load side position error is used, respectively.

By minimizing (5.9), it is then possible to find an optimal input, $\mathbf{U}_{c,j+1}$ to the system, (5.5) and (5.6) with respect to the criterion (5.9).

5.4 Motor Side Iterative Learning Controller

To obtain the motor side iterative learning controller, (5.9) is rewritten using the motor side position error as

$$\mathbf{J}_{m,j+1} = \mathbf{E}_{m,j+1}^T \mathbf{W}_e \mathbf{E}_{m,j+1} + \mathbf{U}_{c,j+1}^T \mathbf{W}_u \mathbf{U}_{c,j+1} \tag{5.10}$$

From (5.10) and (5.5), it follows that the derivative of $\mathbf{J}_{m,j+1}$ with respect to $\mathbf{U}_{c,j+1}$ is

$$\begin{aligned} \frac{\partial \mathbf{J}_{m,j+1}}{\partial \mathbf{U}_{c,j+1}} &= \frac{\partial}{\partial \mathbf{U}_{c,j+1}}((\mathbf{R}_m - \mathbf{Y}_{m,j})^T \mathbf{W}_e (\mathbf{R}_m - \mathbf{Y}_{m,j})) + \frac{\partial}{\partial \mathbf{U}_{c,j+1}}(\mathbf{U}_{c,j+1}^T \mathbf{W}_u \mathbf{U}_{c,j+1}) \\ &= \mathbf{T}_m^T \mathbf{W}_e (\mathbf{C}_{ff1} - \mathbf{T}_{mr}) \mathbf{R}_\ell - \mathbf{T}_m^T \mathbf{W}_e \mathbf{T}_m \mathbf{D}_{j+1} - \mathbf{T}_m^T \mathbf{W}_e \mathbf{T}_m \mathbf{U}_{ff} \\ &\quad + (\mathbf{T}_m^T \mathbf{W}_e \mathbf{T}_m + \mathbf{W}_u) \mathbf{U}_{c,j+1} \end{aligned} \tag{5.11}$$

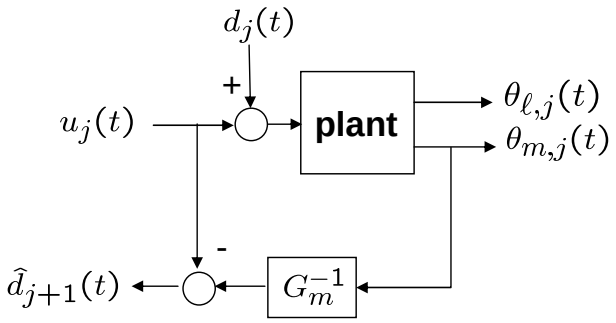


Figure 5.2: Disturbance observer structure - Motor side ILC

The optimal input, $\mathbf{U}^*_{c,j+1}$, with respect to the criterion (5.10) can be obtained by setting $\frac{\partial \mathbf{J}_{m,j+1}}{\partial \mathbf{U}_{c,j+1}} = 0$ and it follows,

$$\mathbf{U}^*_{c,j+1} = \Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m\mathbf{D}_{j+1} + \Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m\mathbf{U}_{ff} + \Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e(\mathbf{T}_{mr} - \mathbf{C}_{ff1})\mathbf{R}_\ell \tag{5.12}$$

where $\Delta = (\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m + \mathbf{W}_u)$

If $\mathbf{W}_u = 0$, i.e., the input energy is not penalized and the feedforward controller is designed such that $\mathbf{U}_{ff} = -\triangle^{-1}\mathbf{T}_m^T\mathbf{W}_m(\mathbf{T}_{mr} - \mathbf{C}_{ff1})\mathbf{R}_\ell$, the updating scheme of the iterative learning controller, (5.12), becomes

$$\mathbf{U}^*_{c,j+1} = \mathbf{D}_{j+1} \tag{5.13}$$

Equation (5.13) represents the intuitive approach for the disturbance rejection problem in Figure 5.1 where the compensation input generated by the learning controller is equal to the input disturbance, $\mathbf{D}_{j+1}$, such that the effect of $\mathbf{D}_{j+1}$ can be exactly canceled. Note that in general, $\mathbf{D}_{j+1}$ is not available when $\mathbf{U}_{c,j+1}$ is calculated. However, if the disturbance, $\mathbf{D}_{j+1}$, does not change dramatically from one iteration to another and a model of the system is available, the disturbance, $\mathbf{D}_{j+1}$, can be replaced by the disturbance estimate, $\hat{\mathbf{D}}_{j+1}$, where $\hat{\mathbf{D}}_{j+1}$ can obtained by a disturbance observer from the previous iteration.

Thus, equation (5.12) can be rewritten by replacing $\mathbf{D}_{j+1}$ with its estimate as

$$\mathbf{U}_{c,j+1} = \Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m\hat{\mathbf{D}}_{j+1} + \Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m\mathbf{U}_{ff} + \Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e(\mathbf{T}_{mr} - \mathbf{C}_{ff1})\mathbf{R}_\ell \tag{5.14}$$

Disturbance observer (DOB) is an approach particularly for handling disturbances in motion control system. It was introduced by Ohnishi (1987) [51] and refined by Umeno and Hori (1991) [63]. The disturbance observer estimates the disturbance, $d(t)$, which is defined as the difference between the actual output and the

output of the nominal model due to the disturbance, $d(t)$, applied to the nominal model. Figure 5.4 shows the structure of the disturbance observer using motor side information. In the figure, the motor position, θ_m, is processed by the inverse of the motor side nominal dynamics of the plant, G_m^{-1}. This results in the disturbance estimate for the motor side learning controller:

$$\hat{d}_{j+1}(t) = G_m^{-1} y_{m,j}(t) - u_j(t) \tag{5.15}$$

If the plant, G_m, has the non-minimum phase zero, the zero phase error tracking (ZPET) method [58] is used to obtain the inverse of the plant, G_m^{-1}. Equation (5.15) can be formulated in the matrix form where

$$\hat{\mathbf{D}}_{j+1} = \mathbf{G}_m^{-1}\mathbf{Y}_{m,j} - \mathbf{U}_j \tag{5.16}$$

Note that even though the inverse of plant is normally not realizable, because the learning controller is applied in the iteration domain, it is now possible to obtain the future time information within one iteration.

By using (5.16) and (5.5), (5.14), i.e., the motor side iterative learning controller, can be obtained as

$$\begin{aligned}\mathbf{U}_{c,j+1} &= \Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m(\mathbf{G}_m^{-1}\mathbf{Y}_{m,j} - \mathbf{U}_j) + \Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m\mathbf{U}_{ff} \\ &\quad +\Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e(\mathbf{T}_{mr} - \mathbf{C}_{ff1})\mathbf{R}_\ell \\ &= -\Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m\mathbf{G}_m^{-1}\mathbf{E}_{m,j} - \Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m(\mathbf{C}\mathbf{T}_m - \mathbf{I})\mathbf{U}_{c,j}\end{aligned} \tag{5.17}$$

with a feedforward controller

$$\mathbf{U}_{ff} = (\mathbf{G}_m^{-1}\mathbf{C}_{ff1})\mathbf{R}_\ell \tag{5.18}$$

Notice that (5.17) can be formed into the standard iterative learning controller form

$$\mathbf{U}_{c,j+1} = \mathbf{Q}_m(\mathbf{U}_{c,j} + \mathbf{L}_m\mathbf{E}_{m,j}) \tag{5.19}$$

where

$$\begin{aligned}\mathbf{Q}_m &= -\Delta^{-1}\mathbf{T}_m^T\mathbf{W}_e\mathbf{T}_m(\mathbf{C}\mathbf{T}_m - \mathbf{I}) \\ \mathbf{L}_m &= [\mathbf{G}_m(\mathbf{C}\mathbf{T}_m - \mathbf{I})]^{-1}\end{aligned}$$

Notice that by using (5.5) and (5.18), (5.19) can be written as

$$\begin{aligned}\mathbf{U}_{c,j+1} &= \mathbf{Q}_m(\mathbf{U}_{c,j} + \mathbf{L}_m(\mathbf{C}_{ff1}\mathbf{R}_\ell - \mathbf{Y}_{m,j})) \\ &= \mathbf{Q}_m(\mathbf{I} - \mathbf{L}_m\mathbf{T}_m)\mathbf{U}_{c,j} \\ &\quad +\mathbf{Q}_m\mathbf{L}_m[(\mathbf{C}_{ff1} - \mathbf{T}_{mr} + \mathbf{T}_m\mathbf{G}_m^{-1}\mathbf{C}_{ff1})\mathbf{R}_\ell + \mathbf{T}_m\mathbf{D}_j]\end{aligned} \tag{5.20}$$

It has been shown in the literature [6] that the motor side iterative learning control system, (5.20), is asymptotically stable if the spectral radius of $\mathbf{Q}_m(\mathbf{I} - \mathbf{L}_m\mathbf{T}_m)$ is smaller than one, i.e.,

$$\rho(\mathbf{Q}_m(\mathbf{I} - \mathbf{L}_m\mathbf{T}_m)) < 1 \tag{5.21}$$

Furthermore, if the maximum singular value is smaller than one, i.e.

$$\overline{\sigma}(\mathbf{Q}_m(\mathbf{I} - \mathbf{L}_m\mathbf{T}_m)) < 1 \tag{5.22}$$

then the iterative learning control system is asymptotically stable, and the motor side learning signal $\mathbf{U}_{c,j}$ converges monotonically.

5.5 Load Side Iterative Learning Controller

The load side iterative learning controller can be obtained in a similar way as the motor side learning controller described in the previous section. The optimization criterion, (5.9), for the load side iterative learning controller can be rewritten using the load side position error, $\mathbf{E}_{\ell,j} = \mathbf{R}_\ell - \mathbf{Y}_{\ell,j}$, as

$$\mathbf{J}_{\ell,j+1} = \mathbf{E}_{\ell,j+1}^T\mathbf{W}_e\mathbf{E}_{\ell,j+1} + \mathbf{U}_{c,j+1}^T\mathbf{W}_u\mathbf{U}_{c,j+1} \tag{5.23}$$

Thus, the optimal input, $\mathbf{U}_{c,j+1}$, with respect to the criterion (5.23) can be obtained as

$$\mathbf{U}_{c,j+1}^* = \Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e\mathbf{T}_\ell\mathbf{D}_{j+1} + \Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e\mathbf{T}_\ell\mathbf{U}_{ff} + \Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e(\mathbf{T}_{\ell r} - \mathbf{I})\mathbf{R}_\ell \tag{5.24}$$

where $\Delta = (\mathbf{T}_\ell^T\mathbf{W}_e\mathbf{T}_\ell + \mathbf{W}_u)$

Notice that for the load side learning controller, the relationship represented in (5.13) can be derived analogously and the disturbance in the $j+1^{th}$ iteration, $\mathbf{D}_{j+1}$, in (5.24) is now replaced by the load side disturbance estimate, $\hat{\mathbf{D}}_{j+1}$.

$$\mathbf{U}_{c,j+1} = \Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e\mathbf{T}_\ell\hat{\mathbf{D}}_{j+1} + \Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e\mathbf{T}_\ell\mathbf{U}_{ff} + \Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e(\mathbf{T}_{\ell r} - \mathbf{I})\mathbf{R}_\ell \tag{5.25}$$

Figure 5.5 shows the disturbance observer structure using load side information. In the figure, the load side position measurement, θ_ℓ, is processed by the inverse of the load side nominal plant, G_ℓ^{-1}. In real applications, however, the load side position measurement, θ_ℓ, is often not available. Thus, a load side position estimation algorithm which uses the load side acceleration measurement, $\ddot{\theta}_\ell$, and the input torque, u, is used to find the load side position estimate, $\hat{\theta}_\ell$, instead of the real load side position measurement, θ_ℓ. The estimation process of $\hat{\theta}_\ell$ will be explained in the

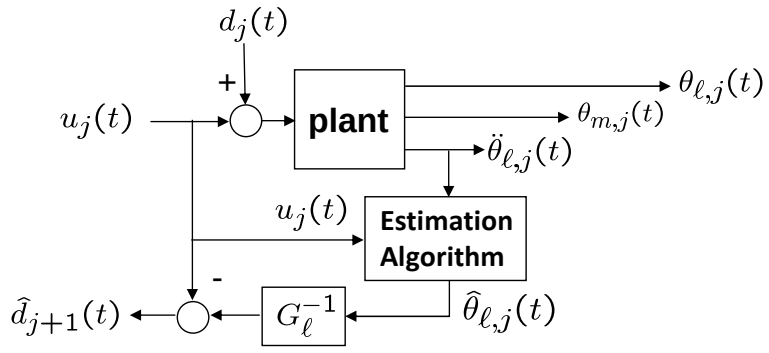


Figure 5.3: Disturbance observer structure - Load side ILC

following section. The resulting load side disturbance estimate can be formulated in the matrix form

$$\hat{\mathbf{D}}_{j+1} = \mathbf{G}_\ell^{-1}\hat{\mathbf{Y}}_{\ell,j} - \mathbf{U}_j \tag{5.26}$$

By using (5.25) and (5.26), the load side iterative learning controller can be obtained as

$$\begin{aligned}\mathbf{U}_{c,j+1} &= \Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e\mathbf{T}_\ell(\mathbf{G}_\ell^{-1}\hat{\mathbf{Y}}_{\ell,j} - \mathbf{U}_j) + \Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e\mathbf{T}_\ell\mathbf{U}_{ff} \\ &\quad +\Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e(\mathbf{T}_{\ell r} - \mathbf{I})\mathbf{R}_\ell \\ &= -\Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_\ell\mathbf{T}_\ell\mathbf{G}_\ell^{-1}\mathbf{E}_{\ell,j} - \Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_\ell\mathbf{T}_\ell(\mathbf{C}\mathbf{T}_m - \mathbf{I})\mathbf{U}_{c,j}\end{aligned} \tag{5.27}$$

with a feedforward controller

$$\mathbf{U}_{ff} = (\mathbf{T}_{\ell r} - \mathbf{C}\mathbf{C}_{ff1})\mathbf{R}_\ell \tag{5.28}$$

Notice that (5.27) can also be written in the standard iterative learning controller form:

$$\mathbf{U}_{c,j+1} = \mathbf{Q}_\ell(\mathbf{U}_{c,j} + \mathbf{L}_\ell\mathbf{E}_{\ell,j}) \tag{5.29}$$

where

$$\begin{aligned}\mathbf{Q}_\ell &= -\Delta^{-1}\mathbf{T}_\ell^T\mathbf{W}_e\mathbf{T}_\ell(\mathbf{C}\mathbf{T}_\ell - \mathbf{I}) \\ \mathbf{L}_\ell &= [\mathbf{G}_\ell(\mathbf{C}\mathbf{T}_\ell - \mathbf{I})]^{-1}\end{aligned}$$

As explained in Section 5.4, the load side iterative learning control system, (5.29), can be shown to be asymptotically stable if the spectral radius of $\mathbf{Q}_\ell(\mathbf{I}-\mathbf{L}_\ell\mathbf{T}_\ell)$ is smaller than one, i.e.

$$\rho(\mathbf{Q}_\ell(\mathbf{I} - \mathbf{L}_\ell\mathbf{T}_\ell)) < 1 \tag{5.30}$$

Furthermore, if the maximum singular value is smaller than one, i.e.

$$\overline{\sigma}(\mathbf{Q}_\ell(\mathbf{I} - \mathbf{L}_\ell\mathbf{T}_\ell)) < 1 \tag{5.31}$$

then the iterative learning control system is asymptotically stable, and the load side learning signal $\mathbf{U}_{c,j}$ converges monotonically.

5.6 Load Side Position Estimation

As mentioned in the previous chapters, precise load side (end-effector) measurements are usually not available in industrial robots due to the cost and assembly issues. Therefore, low-cost MEMS sensors, such as gyroscopes and accelerometers are widely used in motion control applications and are integrated on the load side for estimation purposes. Problems such as non-negligible biases, limited bandwidth, and noises from inexpensive sensors, however, can set restrictions on the direct utilization of the load side sensor signals.

The single-joint indirect drive train system is equipped with two accelerometers on the payload as described in Section 2.2. A Kalman filter that makes use of the acceleration measurement was introduced in Section 4.3. It was used for eliminating the bias in the acceleration measurement by processing the estimated acceleration signal through a low pass filter. The method was shown to be sufficient for the purpose of eliminating the bias effect. To accurately provide the load side position estimate, however, the Kalman filter in Section 4.3, which only considers the plant dynamics is not adequate due to the apparent sensor dynamics. In order to obtain the precise load side position estimate, the sensor dynamics have to be considered in the estimation process.

The sensor dynamics that includes the effect of bias and noise can be described in general as

$$\begin{aligned} \dot{z}_{bias}(t) &= n_b(t) \\ z_m(t) &= z_r(t) + z_{bias}(t) + n_z(t) \end{aligned} \tag{5.32}$$

where $z_{bias}(t)$ is the bias in the sensor measurement which is modeled and governed by the noise, $n_b(t)$. $z_m(t)$ is the measured signal obtained at the sensor output and $z_r(t)$ is the corresponding actual signal at the system output. $n_z(t)$ is the sensor noise.

The state space equation for the single-joint indirect drive train system, (2.3), with the consideration of sensor dynamics, (5.32), can be written as

$$\begin{aligned} \dot{\mathbf{f}}(t) &= \mathbf{A}_{aug}\mathbf{f}(t) + \mathbf{B}_{aug,u}u(t) + \mathbf{B}_{aug,w}\mathbf{w}_c(t) \\ \mathbf{h}(t) &= \mathbf{C}_{aug}\mathbf{f}(t) + \mathbf{v}_c(t) \end{aligned} \tag{5.33}$$

where

$$\begin{aligned}
\mathbf{A}_{aug} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \quad \mathbf{B}_{aug,u} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}, \quad \mathbf{w}_c(t) = \begin{bmatrix} n_u(t) \\ n_b(t) \end{bmatrix}, \\
\mathbf{v}_c(t) &= \begin{bmatrix} n_{sm}(t) \\ n_{a\ell}(t) \end{bmatrix}, \quad \mathbf{h}(t) = \begin{bmatrix} \overline{\theta}_m(t) \\ \overline{\ddot{\theta}}_\ell(t) \end{bmatrix}, \quad \mathbf{f}(t) = \begin{bmatrix} x(t) \\ \overline{\ddot{\theta}}_{\ell,bias}(t) \end{bmatrix}, \\
\mathbf{C}_{aug} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{k_j}{NJ_\ell} & \frac{d_j}{NJ_\ell} & -\frac{k_j}{J_\ell} & -\frac{d_j+d_\ell}{J_\ell} & 1 \end{bmatrix}, \ \mathbf{B}_{aug,w} = \begin{bmatrix} 0 & \frac{1}{J_m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T
\end{aligned}$$

where $\mathbf{A}$, $\mathbf{B}$, x, and the entries in the matrices $\mathbf{B}_{aug,w}$ and $\mathbf{C}_{aug}$ are the same as those in (2.4). $\overline{\theta}_m(t)$, and $\overline{\ddot{\theta}}_\ell(t)$ are the measured motor side position and load side acceleration, respectively. $\overline{\ddot{\theta}}_{\ell,bias}(t)$ is the bias in the acceleration measurement. $n_u(t)$ and $n_b(t)$ are respectively the noise of the motor torque, $u(t)$, and the accelerometer bias, $\overline{\ddot{\theta}}_{\ell,bias}(t)$. $n_{sm}(t)$ and $n_{a\ell}(t)$ are the measurement noise of the motor side position, $\theta_m(t)$, and the load side acceleration, $\ddot{\theta}_\ell(t)$. $\mathbf{w}_c(t) \sim N(0, \mathbf{Q}_c)$ and $\mathbf{v}_c(t) \sim N(0, \mathbf{R}_c)$ are the process and measurement noise, respectively.

Introduce a sampling time, $\triangle t$, as a zero order hold on the input side of the plant, (5.33) can be written in the discrete-time state space form as

$$\begin{aligned}
\mathbf{f}(k+1) &= \mathbf{A}_{augz}\mathbf{f}(k) + \mathbf{B}_{augz,u}u(k) + \mathbf{B}_{augz,w}\mathbf{w}(k) \\
\mathbf{h}(k) &= \mathbf{C}_{augz}\mathbf{f}(k) + \mathbf{v}(k)
\end{aligned} \tag{5.34}$$

where

$$\begin{aligned}
\mathbf{A}_{augz} &= e^{\mathbf{A}_{aug}\triangle t}, \qquad \mathbf{B}_{augz,u} = \int_0^{\triangle t} e^{\mathbf{A}_{aug}\eta}\mathbf{B}_{aug,u}d\eta \\
\mathbf{B}_{augz,w} &= \int_0^{\triangle t} e^{\mathbf{A}_{aug}\eta}\mathbf{B}_{aug,w}d\eta
\end{aligned}$$

$\mathbf{w}(k)$ and $\mathbf{v}(k)$ are the discrete time process and measurement noise where $\mathbf{w}(k) \sim N(0, \mathbf{Q})$ and $\mathbf{v}(k) \sim N(0, \mathbf{R})$. Notice that the relationship of the variance matrices, $\mathbf{Q}$, $\mathbf{Q}_c$, $\mathbf{R}$, and $\mathbf{R}_c$, are approximately $\mathbf{Q} = \mathbf{Q}_c\triangle t$ and $\mathbf{R} = \mathbf{R}_c/\triangle t$ as shown in [56]. For the system shown in (5.34), a Kalman filter that provides the system state estimates, $\hat{\mathbf{f}}$, can be found as

$$\hat{\mathbf{f}}(k+1|k+1) = \hat{\mathbf{f}}(k+1|k) + \mathbf{K}(k+1)[\mathbf{h}(k+1) - \mathbf{C}_{augz}\hat{\mathbf{f}}(k+1|k)] \tag{5.35}$$

where

$$\begin{aligned}
\hat{\mathbf{f}}(k+1|k) &= \mathbf{A}_{augz}\hat{\mathbf{f}}(k|k) + \mathbf{B}_{augz,u}u(k) \\
\mathbf{M}(k+1) &= \mathbf{A}_{augz}\mathbf{P}(k)\mathbf{A}_{augz}^T + \mathbf{B}_{augz,w}\mathbf{Q}\mathbf{B}_{augz,w}^T \\
\mathbf{K}(k+1) &= \mathbf{M}(k+1)\mathbf{C}_{augz}^T[\mathbf{C}_{augz}\mathbf{M}(k+1)\mathbf{C}_{augz}^T + \mathbf{R}]^{-1} \\
\mathbf{P}(k+1) &= [\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}_{augz}]\mathbf{M}(k+1)
\end{aligned}$$

Table 5.1: The noise variance used in experiments

$\mathbf{Q}(n_u)$	$\mathbf{R}(n_{sm})$	$\mathbf{R}(n_{a\ell})$	$\mathbf{Q}(n_b)$
10^{-4}	8.2247×10^{-9}	3.2841×10^{-3}	10^{-5}

Notice that $\mathbf{K}(k)$ is the Kalman filter gain. $\mathbf{M}(k)$ and $\mathbf{P}(k)$ are the a priori and the a posteriori estimate error covariance, respectively. Notice that $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are often interpreted as fictitious noise terms and their variance can be adjusted to assign a reasonable set of closed loop eigenvalues to the estimator. In practice, they are chosen such that the estimator dynamics is five to ten times faster than that of the controller or fast enough to suppress the effect of perturbations [10].

5.7 Experimental Study

5.7.1 Experimental Setup

The proposed iterative learning controllers are applied to the single-joint indirect drive train setup described in Section 2.2. Notice that the experimental setup is quipped with a load side encoder. The load side position measurement, however, is used only for evaluating the performance of the proposed controllers. The reference load side trajectory is shown in Figure 5.4. It is obtained by the 4^{th} order trajectory generation method explained in Section 3.4.3. It can be seen from the figure that, the payload of the single-joint indirect drive train moves from the zero position to the desired position with an acceleration phase, a constant speed phase and a deceleration phase and then returns to the zero position in the same manner in 3.4 seconds. The controller gains, $[K_p,\ K_v,\ K_i]$, of the modified PID controller, C, are chosen to be [30, 0.3, 1] for the experiments. The controller gains, $[K_{f1},\ K_{f2}]$, of the feedforward controller, C_{ff1}, are chosen to be $[10^{-5},\ 0.0275]$.

The noise variance used in the experiments are shown in Table 5.1. The noise of the motor side position, θ_m, is bounded by the encoder resolution $\delta\theta_m$. This gives the approximate output noise variance $\mathbf{R}(n_m) = \delta\theta_m^2/12$. The noise variance for the load side accelerometer output, $\ddot{\theta}_\ell$, can be obtained by zero-acceleration experiment, which gives $\mathbf{R}(n_{a\ell}) = 3.2841 \times 10^{-3}(rad/sec^2)^2$. Since no torque measurement is available in this experimental setup, the noise variance of the motor torque output, $\mathbf{Q}(n_u)$, can not be determined experimentally, and thus could be designed for the controller performance. Also, the noise variance of the measurement bias, $\mathbf{Q}(n_b)$, can be used as the design parameter [10].

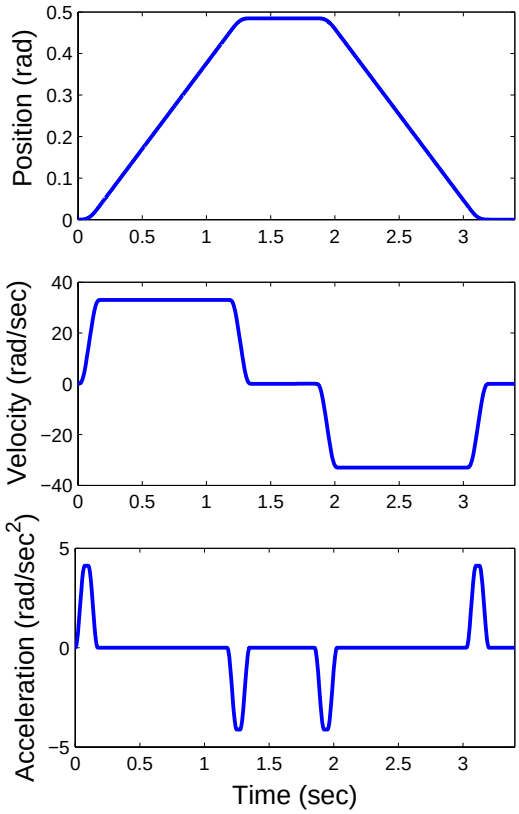


Figure 5.4: Desired load side trajectory

5.7.2 Experimental Results

Load side position estimation

A position estimation algorithm was proposed in Section 5.6. Notice that if the estimation algorithm is not available, the load side position, θ_ℓ, can be approximated based on the system dynamics model using the load side acceleration measurement, $\overline{\ddot{\theta}}_\ell$, and motor side position measurements, $\overline{\theta}_m$. From (2.3), the load side dynamics of the single-joint indirect drive train is

$$J_\ell \ddot{\theta}_\ell + d_\ell \dot{\theta}_\ell = k_j \left(\frac{\theta_m}{N} - \theta_\ell \right) + d_j \left(\frac{\dot{\theta}_m}{N} - \dot{\theta}_\ell \right)$$

Assuming that the damping terms in the above equation can be ignored, the load side position can be estimated as

$$\hat{\theta}_{\ell wo} = \frac{\bar{\theta}_m}{N} - \frac{J_\ell}{k_j} \bar{\ddot{\theta}}_\ell \tag{5.36}$$

It follows that the load side estimation error for (5.36) can be found as

$$\hat{e}_{\ell wo} = \bar{\theta}_\ell - \hat{\theta}_{\ell wo} \tag{5.37}$$

where $\overline{\theta}_\ell$ is the measured load side position.

The solid and dashed lines in Figure 5.5 show the load side position estimation error, $\hat{e}_{\ell w}$, and, $\hat{e}_{\ell wo}$, respectively. $\hat{e}_{\ell wo}$ is defined in (5.37) and $\hat{e}_{\ell w} = \bar{\theta}_\ell - \hat{\theta}_{\ell w}$. $\hat{\theta}_{\ell w}$ is the load side position estimate obtained by the estimation algorithm proposed in Section 5.6. It can be seen from the figure that the estimated load side position $\hat{\theta}_{\ell wo}$ in (5.36) suffers from the bias and the noises of the acceleration measurement, $\overline{\ddot{\theta}}_\ell$ while the load side position estimate, $\hat{\theta}_{\ell w}$, obtained in Section 5.6 provides a more satisfactory estimation. Figure 5.6 shows the bias estimation of the load side acceleration measurement. It is clear that the estimation method proposed in Section 5.6 is superior compare to the method in (5.36) and it is effective to estimate the measurement bias and thus reduce the bias effects in the utilization of sensor signals.

Motor side iterative learning controller

Figure 5.7 shows the results of the motor side learning controller after 10 iterations. The weighting matrices in (5.10) are chosen to be $\mathbf{W}_u = 10^{-2} \cdot \mathbf{I}$ and $\mathbf{W}_e = 10^2 \cdot \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. Figure 5.7(a) shows the motor position error where the dashed and solid lines represent the time responses before and after applying the motor side iterative learning controller, respectively. Figure 5.7(b)

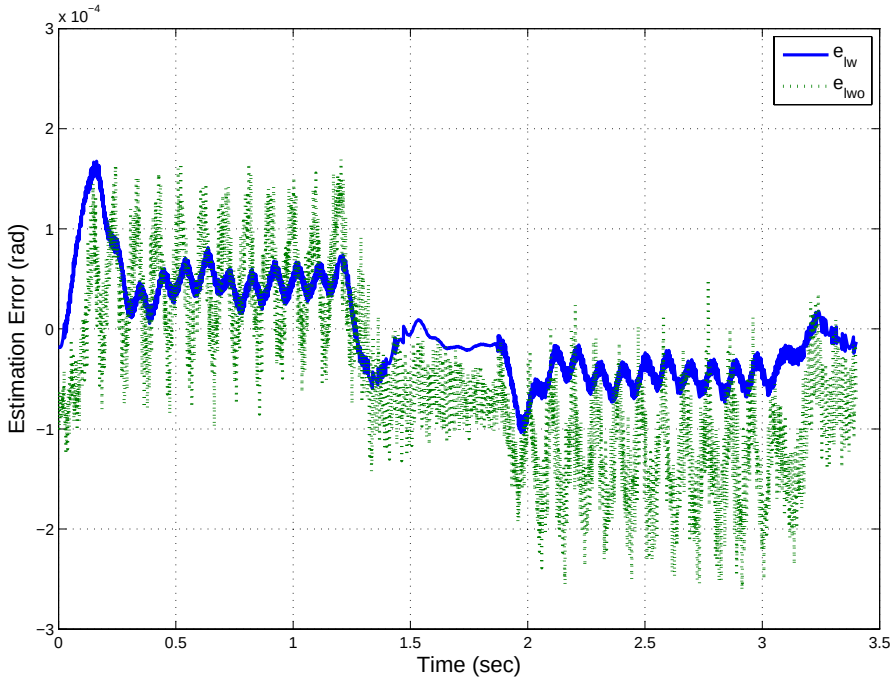


Figure 5.5: Comparison of the load side position estimation error

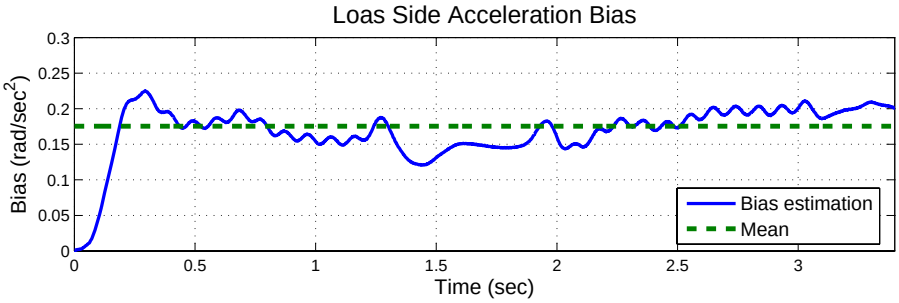
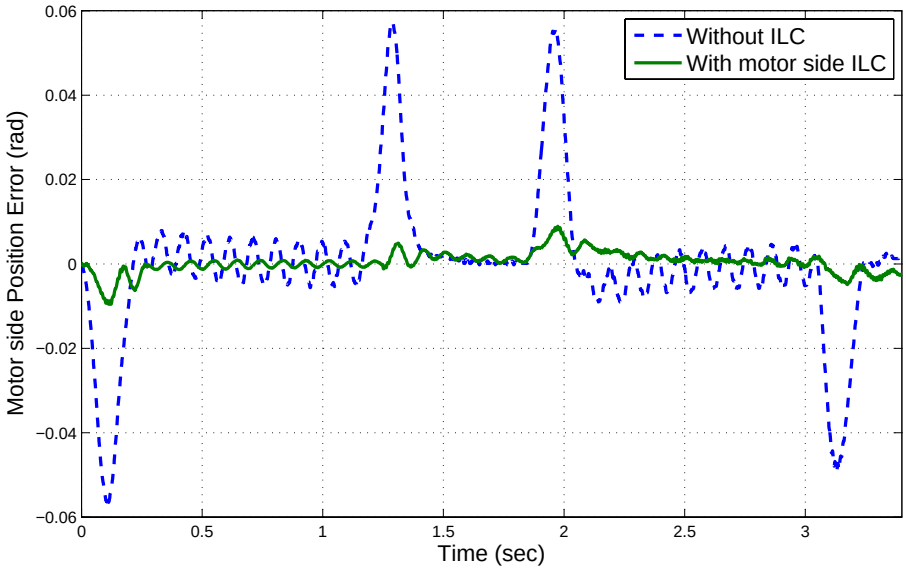
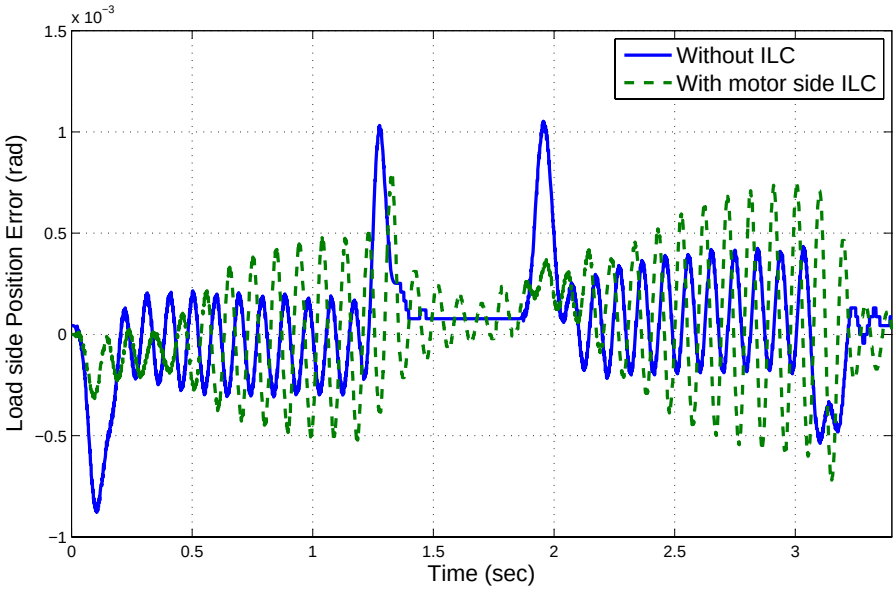


Figure 5.6: Bias estimation of the acceleration measurement



(a) Motor side position error before and after



(b) Load side position error before and after

Figure 5.7: Motor side iterative learning controller results

shows the load side position error where the solid and dashed lines represent the time responses before and after applying the motor side iterative learning controller, respectively. It can be observed from Figure 5.7(a) that the motor side learning controller reduces the motor side vibrations effectively. The load side position error as shown in Figure 5.7(b), however, was deteriorated by the motor side learning controller. The load side tracking error during the constant speed phase of the trajectory was increased with a phase error. It is shown in the figure that with the presence of the disturbance, the perfect disturbance rejection on the motor side will result in a nonzero error on the load side. Since the load side iterative learning controller directly learns the error from the load side, it is expected that the load side learning controller will perform better than the motor side one in terms of load side tracking performance.

Load side iterative learning controller

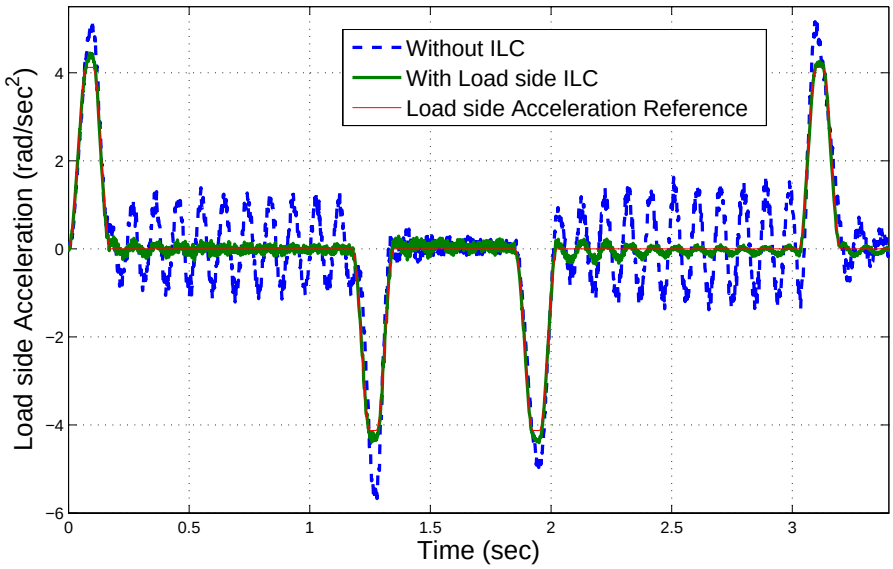
Figure 5.8 shows the results of the load side iterative learning controller after 10 iterations with the weighting matrices, $\mathbf{W}_u = 10^{-2} \cdot \mathbf{I}$ and $\mathbf{W}_e = 10^5 \cdot \mathbf{I}$. Figure 5.8(a) shows the load side acceleration responses where the dashed and solid line represent the responses before and after applying the load side learning controller, respectively. Figure 5.8(b) shows the load side position error responses where the dashed and solid line represent the responses before and after applying the load side iterative learning controller, respectively. The load side learning controller reduces the vibrations on the load side effectively not only in the acceleration and deceleration phase but also the constant speed phase. The experimental results clearly show that the load side learning controller is more effective to improve the load side performance.

Comparison

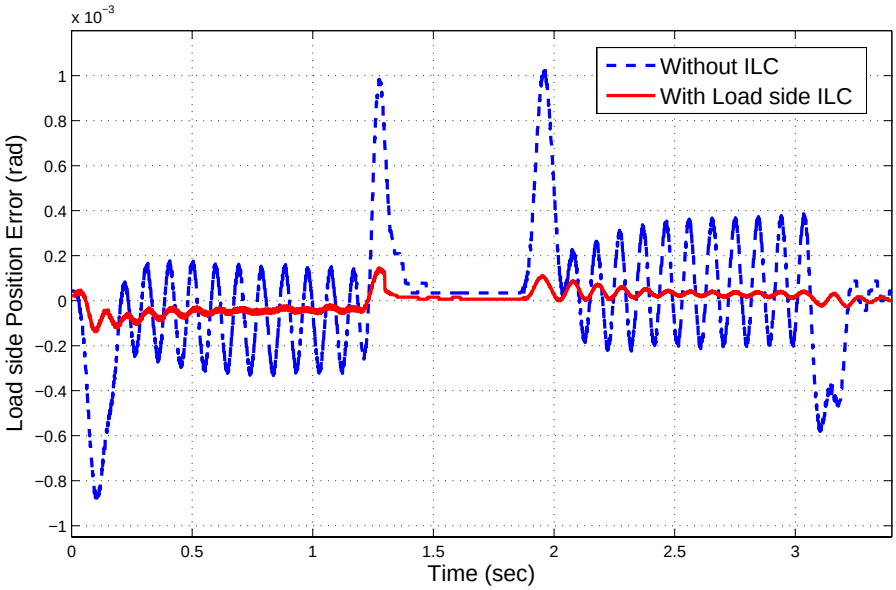
The performance of the learning controllers is further evaluated in the iteration domain by the root-mean-square (RMS) values of the error signals. Notice that the RMS of the error signal is computed as

$$e_{RMS} = \left(\sum_k e^2(k) \right)^{\frac{1}{2}}$$

Figure 5.9 shows the convergence of the load side position error signals where the the circle ($\circ$) and the square ($\square$) represent the error from the load side and the motor side iterative learning controller for each iteration, respectively. It can be seen from the figure that both motor side and load side iterative learning controllers



(a) Load side acceleration responses before and after



(b) Load position error before and after

Figure 5.8: Load side iterative learning controller results

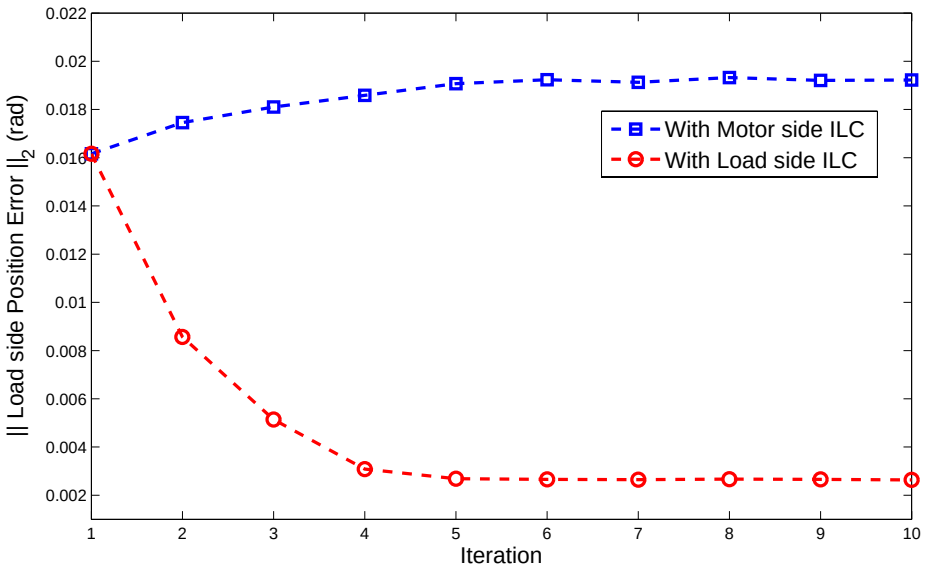


Figure 5.9: Comparison of motor side and load side iterative learning controllers

converge in approximately five iterations. The motor side learning controller, however, amplifies the load side position error while the load side learning controller reduces the position error effectively.

5.8 Summary

An optimization-based iterative learning control scheme for the purpose of disturbance rejection was proposed in this chapter. Two different learning controllers, the motor side and load side learning controller, designed based on different error signal, namely the motor side error, and the load side error, were obtained. A load side position estimation algorithm based on Kalman filtering was proposed and the load side position estimate was used in the load side learning controller. The effectiveness of the estimation scheme was shown with the experimental result. Both learning controllers show good convergence property. However, it was shown that the load side iterative learning controller demonstrates much more substantial reduction of the load side vibration than the motor side learning controller when the load side performance is of interest.

Chapter 6

Identification of Dynamics of a Robot Manipulator

6.1 Introduction

An accurate model of the robot dynamics is an essential prerequisite for many research topics. An accurate model is important not only for the controller design, but also for the development of better or new functionalities, such as collision detection, sensorless force control for assembly, and sensorless lead through teaching. System identification also provides an opportunity to gauge the hardware and software capability prior to controller implementation.

In practice, a SISO system and friction identification are performed to obtain estimates for the system and friction parameters, respectively. Robot manipulators, however, are highly nonlinear, coupled multivariable control systems and unfortunately, SISO system identification does not incorporate this coupling effect. Consequently, SISO sytem identification estimations are not always sufficient to capture the complex dynamics of robot manipulators. As a result, complementary methods such as gain scheduling method, automatic gain tuning method, and adaptive control method are usually applied to improve system performance. If the multi-dimensional system model is identified precisely, it is expected that a MIMO controller can be designed to give better performance over the whole workspace. In addition, it will enable more realistic simulations to evaluate control performance before verifying with experiments. Therefore, a system identification method from the standpoint of control system design is motivated.

The development of the robust control techniques ensures the achievement of stability and specific performance in the presence of plant uncertainties [15]. Note that nearly all robust control design methods require explicit worst-case bounds on the plant uncertainties. A mathematically convenient way to formalize the problem of robust control is to describe the system by a nominal model with a bounded

model error. Hence a good system model is a necessary prerequisite for robust controller design.

The remainder of this chapter is organized as follows. Section 6.2 will first introduce the description of hardware setup. Next, Section 6.3 will discusses the single-input-single-output (SISO) system identification of the M-16*i*B industrial robot provided by FANUC Ltd. Finally, the multi-input-multi-output (MIMO) system identification will be discussed in Section 6.4. The approaches, experimental setup, data processing procedures, and results are summarized in each section.

6.2 Robot Instrumentation

6.2.1 Hardware Setup

Figure 6.1: M-16*i*B FANUC robot

Figure 6.1 shows the M-16*i*B industrial robot used in the current research. The M-16*i*B robot is equipped motor encoders, which will enable motor side data acquisition. Additionally, an accelerometer (Kistler, Type 8330A3)[38] is attached on the payload to measure load side acceleration. And finally, while not used for the SISO system identification, a three-dimensional position measuring system, CompuGauge 3D[13], is also available. It is possible to synchronize the time data with the end-effector position data.

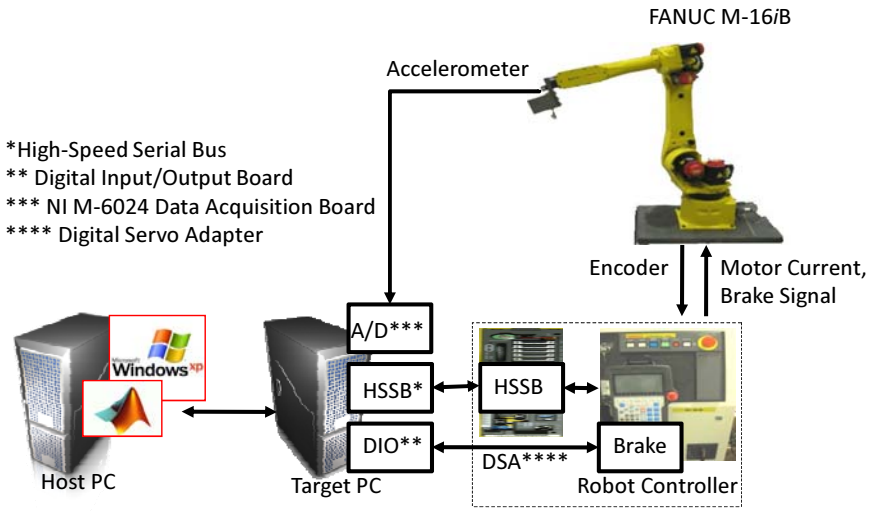


Figure 6.2: Hardware setup

Robot Controller

The commercial M-16*i*B robot controller is capable of position and velocity feedback control. While it is possible to move the robot along any desired trajectory using the provided controller, it does not allow for flexibilities such as adjusting the controller parameters or algorithm. As a result, MATLAB is used to design the control algorithms and a digital to analog servo adapter (DSA) is used to transfer the information to the robot.

By using DSA, the control signal (i.e. motor torque command) is generated by the target computer and is then converted to output current by the robot controller. Although the robot controller is not directly generating signals to control the robot, it is still used to amplify the output from the target computer. Additionally, the robot controller is also used to activate the motor brakes in the robot. Notice that the brake function is turned on and off directly with a digital input/output (DIO) board installed on the target computer.

Real-Time System

Commercial programs that include control toolboxes, such as MATLAB [44] and LABVIEW [46], provide engineers with various advantages. For example, graphical user interfaces (GUI) allow users to design control algorithms in a more intuitive and convenient manner. Furthermore, various other toolboxes, such as System Identification toolbox, Controller Design toolbox, and Signal Processing toolbox provide many useful functions to help users quickly analyze control systems. For

SISO system identification, MATLAB installed on a Windows platform is used to design and implement control algorithms.

In real-time systems, it is critical that hardware performance is not hindered by software computational limitations. Thus, it is important to ensure that the software runs as quickly and consistently as possible. A Windows platform, however, does not guarantee this sort of real-time environment due to unexpected interferences from virus scanning software and other event logging processes.

To overcome these problems, two computers are used for the real-time control system setup. A host computer operated the Windows platform and allowed users to utilize the conveniences of MATLAB. Another computer, called the target computer, is used to process the generated signals and to interface with the robot. The target computer ran an environment based on the XPCtarget system to ensure real-time processing. The current minimum sampling time for XPCtarget is 0.5 msec. Additionally, the robot motor positions, motor velocities, and motor currents are accessible to the target computer through a high-speed serial bus (HSSB) interface. And finally, the load side acceleration information is also available to the target computer through a data acquisition board manufactured by National Instrument.

The host computer and target computer are connected via Ethernet. The control algorithms is designed and compiled on the host computer and then transferred to the target computer. Once the target computer begins running the control algorithm, the connection between the two computer is automatically disengaged, thus allowing the target computer to run without any interference from the host computer. Additional hardware information about the accelerometer mounting and payload design can be found in [66, 68].

6.2.2 Software Setup

In order to expedite the system identification process, a customized GUI is designed in MATLAB as shown in Fig. 6.3. The GUI allows the user to quickly specify system parameters, identify experimental conditions, and analyze collected experimental data. More specifically, the GUI allows users to quickly set up the proper MATLAB work environment, load the desired system model, select the type of experiment to perform as well as configure the experiment parameters, set controller parameters, and generate frequency response plots for the experimental measurements.

6.3 SISO System Identification

6.3.1 System Modeling

For SISO system identification, each joint is individually actuated. When a joint is being actuated, the remaining five joints are assumed to be rigid, hence the joints

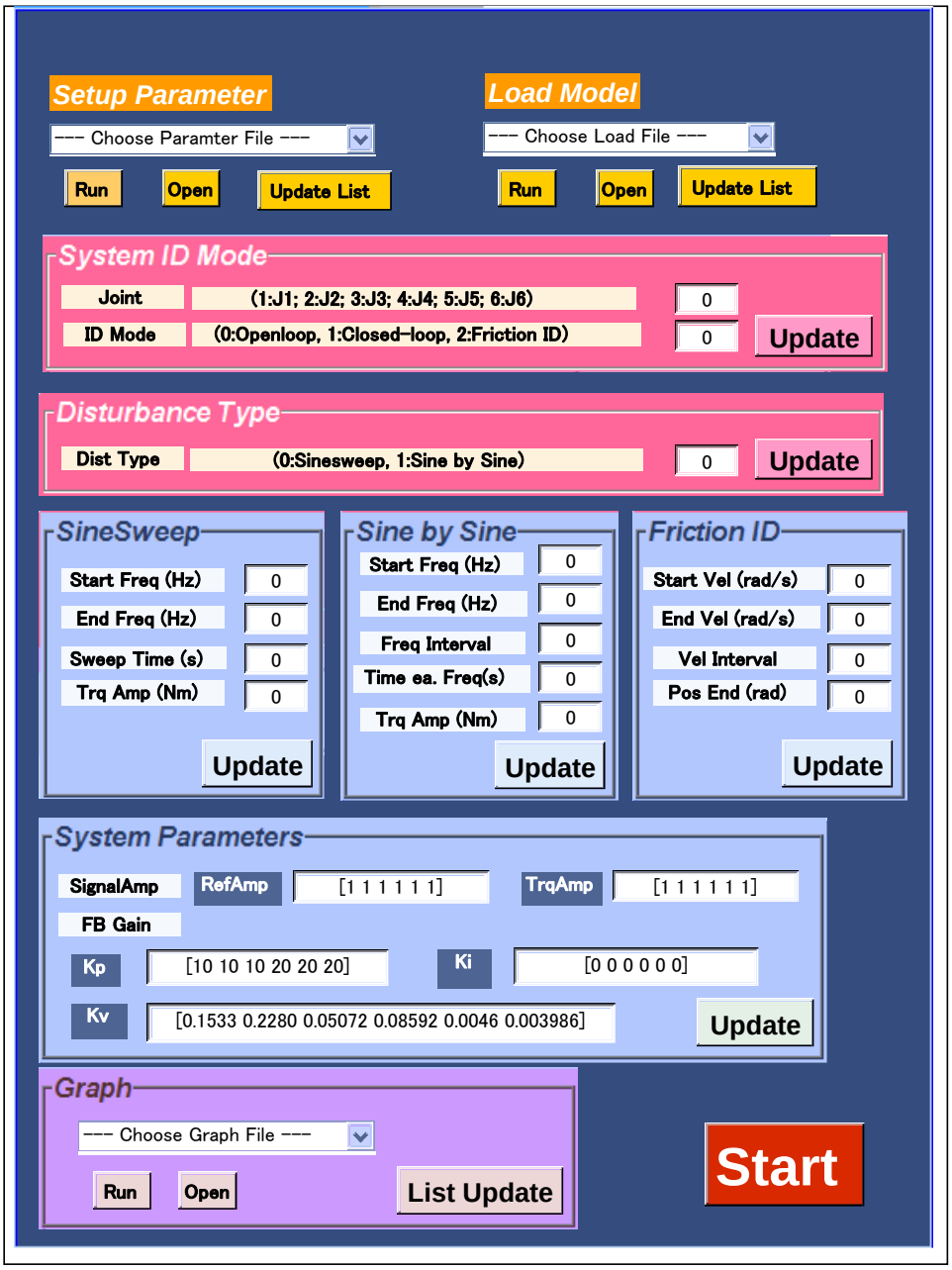


Figure 6.3: MATLAB GUI

are assumed to be decoupled. As a result, it is intuitive to model each joint as a two-inertia mechanism discussed in Section 2.3.

The transfer functions from the torque command, u, to the motor velocity, $\dot{\theta}_m$, and the load side acceleration, $\ddot{\theta}_\ell$, were obtained in Section 2.4:

$$\begin{aligned}\frac{\dot{\theta}_m(s)}{u(s)} &= \frac{J_\ell s^2 + (d_j + d_\ell)s + k_j}{J_m J_\ell s^3 + J_d s^2 + J_k s + k_j(d_m + \frac{d_\ell}{N^2})} \\ \frac{\ddot{\theta}_\ell(s)}{u(s)} &= \frac{d_j s^2 + k_j s}{N[J_m J_\ell s^3 + J_d s^2 + J_k s + k_j(d_m + \frac{d_\ell}{N^2})]}\end{aligned}$$

where

$$\begin{aligned}J_d &= J_m(d_j + d_\ell) + J_\ell\left(\frac{d_j}{N^2} + d_m\right) \\ J_k &= J_m k_j + \frac{J_\ell k_j}{N^2} + (d_j + d_\ell)d_m + \frac{d_j d_\ell}{N^2}\end{aligned}$$

If the system is lightly damped, the damping values d_m, d_j, and d_ℓ can be neglected. By using this simplification, the anti-resonant frequency, ω_{ar}, for the system in (2.5) is approximately:

$$\omega_{ar} = \sqrt{\frac{k_j}{J_\ell}} \tag{6.1}$$

Similarly, the approximate resonant frequency for (2.5) is:

$$\omega_r = \sqrt{\frac{k_j}{J_\ell} + \frac{k_j}{J_m N^2}} \tag{6.2}$$

These approximations are used to obtain the initial estimations for the plant parameters k_j and J_ℓ in the SISO system identification process.

It is important to note that while the joint flexibility parameters, k_j and d_j, are commonly used to capture the motor side to load side dynamics in Rotor-Vector(RV) reducers and harmonic drives, these parameters may not be ideal or even sufficient for modeling other devices. In the M-16*i*B robot, it is known that J_1, J_2, an J_3 use RV reducers and that J_6 uses a harmonic drive. The motor mechanisms driving J_4 and J_5 are known to be customized FANUC designs. But for lack of additional detail regarding these designs, J_4 and J_5 are also modeled using the two-inertia model for the SISO system identification process.

6.3.2 System Identification Approach

Posture Selection

To ensure that the two-inertia model is valid, the robot joint motions must be decoupled from each other during the system identification process. As a result,

Table 6.1: Joint angles for SISO system identification

	Joint Angles (Deg)					
Joint ID	J_1	J_2	J_3	J_4	J_5	J_6
J_1 ID	45	90	0	0	0	90
J_2 ID	0	0	90	90	0	-90
J_3 ID	45	90	90	90	0	-90
J_4 ID	45	90	90	0	90	-90
J_5 ID	45	0	0	90	0	0
J_6 ID	45	0	0	0	90	0

the robot postures for the system identification process are selected to produce the largest inertia with respect to the moving joint while trying to reduce coupling motions in between joints. The joint angles used for each experiment can be found in Table 6.1 and the robot postures are visually shown in Figure 6.4.

Identification Procedure

Figure 6.5 shows the system identification procedure. Since the system is represented using a linear time-invariant (LTI) model, LTI techniques and assumptions can be used to perform the SISO system identification. The system's response to a certain frequency can be obtained by harmonically driving the system at that specific frequency(Figure 6.5(a)). The excitation frequency can be then varied such that the system dynamics are captured over a broad frequency range.

Once the system output measurements are recorded, the frequency response is then calculated by finding the magnitude and phase at each particular frequency. After a plot of the frequency response is obtained, a least square fit is used to empirically obtain the system parameters(Figure 6.5(b)). In the system identification process, the unknown parameters are: J_m, J_ℓ, d_m, d_ℓ, d_j, and k_j. Note that the load side inertia, J_ℓ, to be identified in this chapter is the total inertia considering the combination effect of J_1 to J_6. Take J_1 for example, as shown in Fig. 6.4(a), J_ℓ is the inertia considering the effects from J_2-J_6. Therefore, J_ℓ obtained from SISO system identification is not directly related to the inertia/mass of each joint. However, the corresponding inertia/mass for each joint can be easily calculated according to the standard robot manipulator textbooks. On the other hand, J_m, d_m, d_ℓ, d_j, and k_j obtained from the SISO identification experiments will be used directly as the joint parameters.

For identifying these parameters or fitting transfer function, a good initial guess for each parameter is important. The gear ratio N is assumed to be exactly known

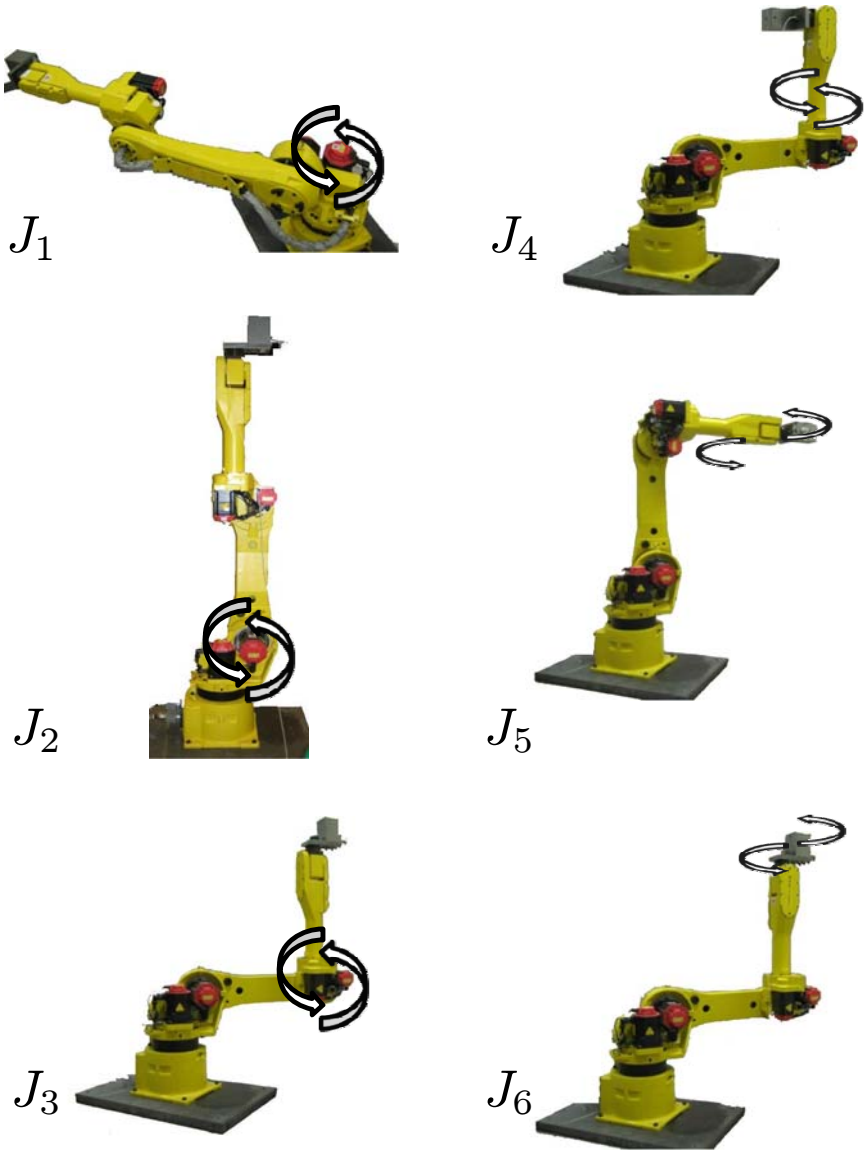
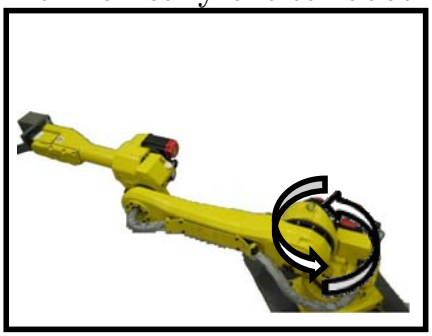


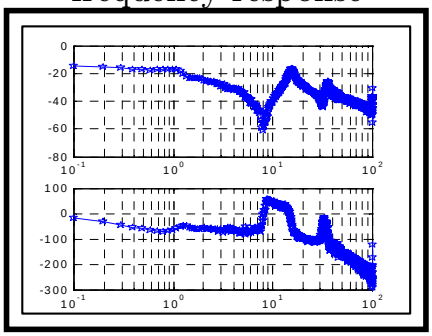
Figure 6.4: Robot postures for SISO system identification

Harmonically excite robot



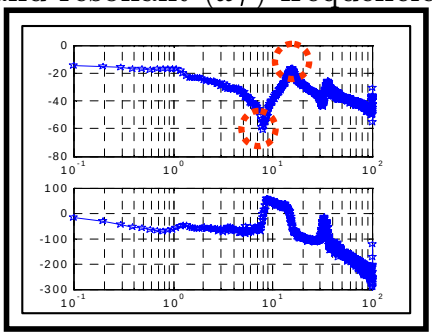
(a)

Acquire data and calculate frequency response



(b)

Locate the anti-resonant (ω_{ar}) and resonant (ω_r) frequencies



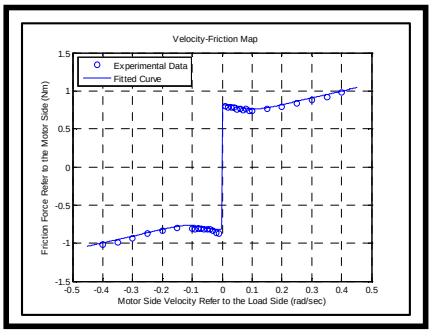
(c)

Obtain initial estimates for J_ℓ and k_j

$$J_\ell = \frac{J_m N^2(\omega_r^2 - \omega_{ar}^2)}{\omega_{ar}^2}$$

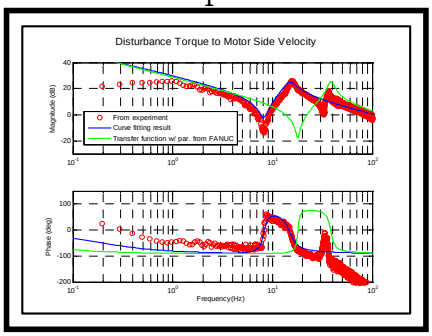
$$k_j = J_m N^2(\omega_r^2 - \omega_{ar}^2)$$

Obtain initial estimates for d_m, d_j, and d_ℓ



(d)

Obtain parameters with nonlinear optimization



(e)

Figure 6.5: System identification flow chart

from the gear catalogue. Based on available technical documentation, a good estimate has been established for J_m for each case in Figure 6.4. By treating J_m and N as knowns and neglecting damping, the initial estimate is obtained for J_ℓ and k_j by using (6.1-6.2) and are expressed as:

$$\begin{aligned} J_\ell &= \frac{J_m N^2(\omega_r^2 - \omega_{ar}^2)}{\omega_{ar}^2} \\ k_j &= J_m N^2(\omega_r^2 - \omega_{ar}^2) \end{aligned}$$

where ω_{ar} and ω_r are read from the calculated frequency response plot as the antiresonance frequency and resonance frequency, respectively. The initial estimates for the damping parameters d_m, d_j, and d_ℓ are obtained through a friction identification process discussed in [25] (Figure 6.5(d)). Once all of the initial system parameters are estimated, constrained non-linear optimization is used to obtain more accurate system parameter estimates (Figure 6.5(e)). The constrained non-linear optimization problem is formulated as follows:

$$\begin{aligned} \min_{\hat{x}} \quad & \sum_{\omega \in \Omega} F(\hat{x}, \omega) \\ \text{s.t.} \quad & A\hat{x} \leq b \end{aligned}$$

where:

$$\begin{aligned} F(\hat{x}, \omega) &= \left|\left|\hat{G}(\hat{x}, \omega) - G_m(\omega)\right|\right|^2 \\ \hat{x} &= \begin{bmatrix} \hat{J}_m & \hat{J}_\ell & \hat{d}_m & \hat{d}_\ell & \hat{d}_j & \hat{k}_j \end{bmatrix}^{\mathrm{T}} \\ A &\in \mathbb{R}^{n \times 6} \\ b &\in \mathbb{R}^{n \times 1} \end{aligned}$$

$\left|\left|\hat{G}(\hat{x}, \omega)\right|\right|$ is the magnitude of the transfer function evaluated with the estimated parameter vector, $\hat{x}$, at frequency ω. $||G_m(\omega)||$ is the magnitude of the measured frequency response at frequency ω. Ω is the set of frequencies at which measurements are taken. Matrix A and vector b are selected to enforce n constraints on the adjustable parameters.

Closed Loop Identification

The experiments for system identification are done in both open and closed loop form. Ideally, an open loop system identification will produce the most accurate results. But due to system instability from gravity effects, open loop identification is not always possible. For J_2 and J_3, only closed loop identifications are performed.

Both open and closed loop identification, however, are initially performed for the remaining four joints to verify the fidelity of the closed loop results.

It is worth mentioning that for both open and closed loop experiments, the joints not being actuated are held rigid by either closed loop feedback or mechanical brakes. Experiments using both methods are conducted and will be discussed in a later section.

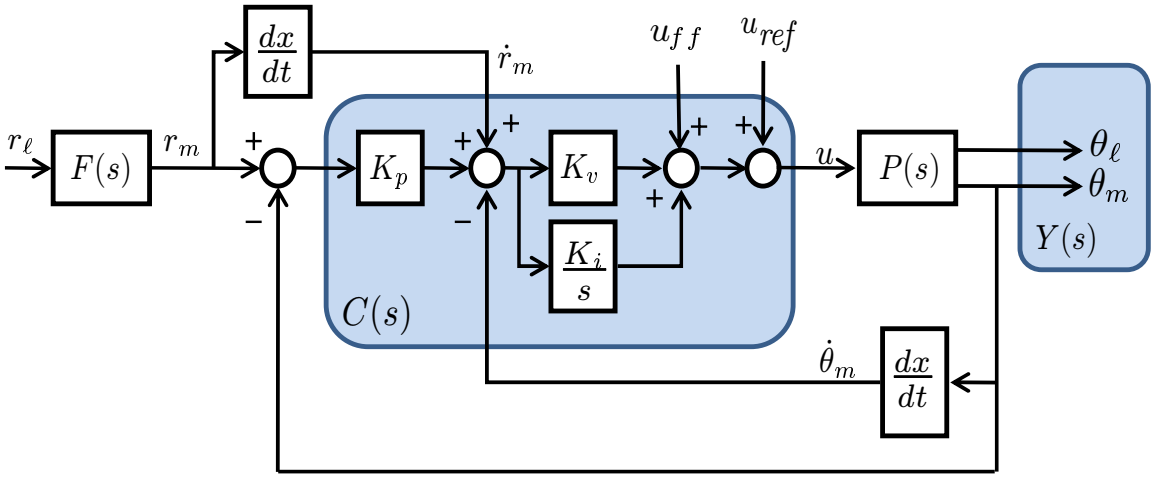


Figure 6.6: Controller structure in closed-loop identification

The controller structure used in the closed loop system identification is a PID controller, $C(s)$, as shown in Figure 6.6. r_ℓ is the load side position reference. $F(s)$ is used to obtain the motor side position reference, r_m. The feedback and feedforward controllers have the same structure as those in Section 3.2. For the purpose of system identification, r_ℓ is held at zero and a sinusoidal input is inserted through u_{ref}. u_{ff} is the feedforward torque for gravity compensation. $P(s)$ denotes the dynamic plant of the two-inertia system.

The gains of $C(s)$ are selected so that $P(s)$ can be accurately identified in a closed loop manner. More specifically, the closed loop transfer function of Figure 6.6 can be expressed as:

$$\frac{Y(s)}{U_{ref}(s)} = \frac{P(s)}{1 + P(s)C(s)} \tag{6.3}$$

The parameters of $C(s)$ are selected such that $P(s)C(s) \ll 1$ at the relevant identification frequencies so that (6.3) is approximately to be $P(s)$ while still maintaining basic control performance, i.e. rejection of unsafe motions.

Table 6.2: Feedback parameters for SISO system identification

Identified Joint		Feedback Controller Gains					
		J_1	J_2	J_3	J_4	J_5	J_6
J_1	K_p	0	10	10	20	20	20
	K_v	0	0.2280	0.05072	0.08592	0.0046	0.003986
J_2	K_p	10	5	10	20	20	20
	K_v	0.1533	0.1	0.05072	0.08592	0.0046	0.003986
J_3	K_p	10	10	2	20	20	20
	K_v	0.1533	0.2280	0.01	0.08592	0.0046	0.003986
J_4	K_p	10	10	10	0	20	20
	K_v	0.1533	0.2280	0.05072	0	0.0046	0.003986
J_5	K_p	10	10	10	20	0	20
	K_v	0.1533	0.2280	0.05072	0.08592	0	0.003986
J_6	K_p	10	10	10	20	20	0
	K_v	0.1533	0.2280	0.05072	0.08592	0.0046	0

6.3.3 Experimental Procedure

For each joint, two different identification experiments are conducted. The sine sweep experiment harmonically excites the joint from 1 Hz to 100 Hz over a 180 sec span. The objective of the sine sweep experiment is to capture an initial estimate of the joint frequency response. Once the sine sweep experiment is completed, the next experiment conducted is the sine by sine test. The sine by sine test is similar to the sine sweep, but instead of continuously increasing the excitation frequency, the sine by sine test excites the joint at a particular frequency for a set amount of time, stops, and then proceeds to the next excitation frequency. There are several motivations for the sine by sine experiment. First, it produces a more accurate frequency response estimate than the sine sweep since the excitation frequency is stationary during each test. Second, it gives the user additional flexibility in setting test parameters. For example, the sine by sine test allows users to excite lower frequencies for longer durations than higher frequencies, which is desirable since data acquisition is slower at lower frequencies.

The two experiments: sine sweep, and sine by sine, on each joint is conducted three times under different condition. For the first set of experiments, the unexcited joints are held in place using closed loop feedback. In the second set of experiments, the unexcited joints are locked in place with the robot's mechanical brakes. The closed loop gains and torque amplitudes used for both sets of experiments are listed in Table 6.2 and Table 6.3. Note that the integral gain, K_i, is set to be zero. The

Table 6.3: Torque amplitudes for SISO system identification

Joint Amplitudes (N·m)	J_1	J_2	J_3	J_4	J_5	J_6
With Payload	1.6	1.6	1.0	0.5	0.2	0.2
Without Payload	1.0	1.0	0.4	0.3	0.15	0.1

closed loop gains and torque amplitudes are the same as before.

6.3.4 Data Processing

Like any experimental measurement, it is impossible to collect data without any form of noise contamination. Signal processing techniques, however, can usually be used to reduce the effects of noise. For example, if a linear system is excited by a sinusoidal input at a single known frequency it is expected to respond only at that frequency. This information can be used to curve fit the measurements such that only one frequency content is present. Figure 6.7 shows a particular measurement. The solid line represents the unprocessed data while the dotted line represents the fitted data. The fitted results are generated by performing a least squares method between the measured results and a sinusoid with the same frequency as the excitation. The least square formulation is as follows:

$$\min_{\hat{x}} \quad \sum_{k \in \Phi} F(\hat{x}, \omega, k) \tag{6.4}$$

where:

$$\begin{aligned} F(\hat{x}, \omega, k) &= \left(\hat{x}_1 \sin(\omega k + \hat{x}_2) - Y_m(\omega, k)\right)^2 \\ \hat{x} &= \begin{bmatrix} \hat{x}_1 & \hat{x}_2 \end{bmatrix} \end{aligned}$$

$\hat{x}$, ω, $Y_m(\omega)$, and k are the parameter vector to be optimized, the excitation frequency, the measured data, and the index respectively. Φ is the set of measured indices. The fitted data allows for the extraction of the amplitude and phase of the system response for a given input. Additionally, the input signal is also fitted with the same least square fit. This fit will simplify the frequency response calculation and will also allow simple phase correction; since in the real-time data logging process, there may be some delay time between the start of the program and the actual execution of the input signals. It is worthwhile to mention that the main contribution of noise is from the inherent joint friction. Note that at higher frequencies (i.e. higher velocities) the friction effects become less evident, hence the least square estimate becomes more accurate at higher frequencies.

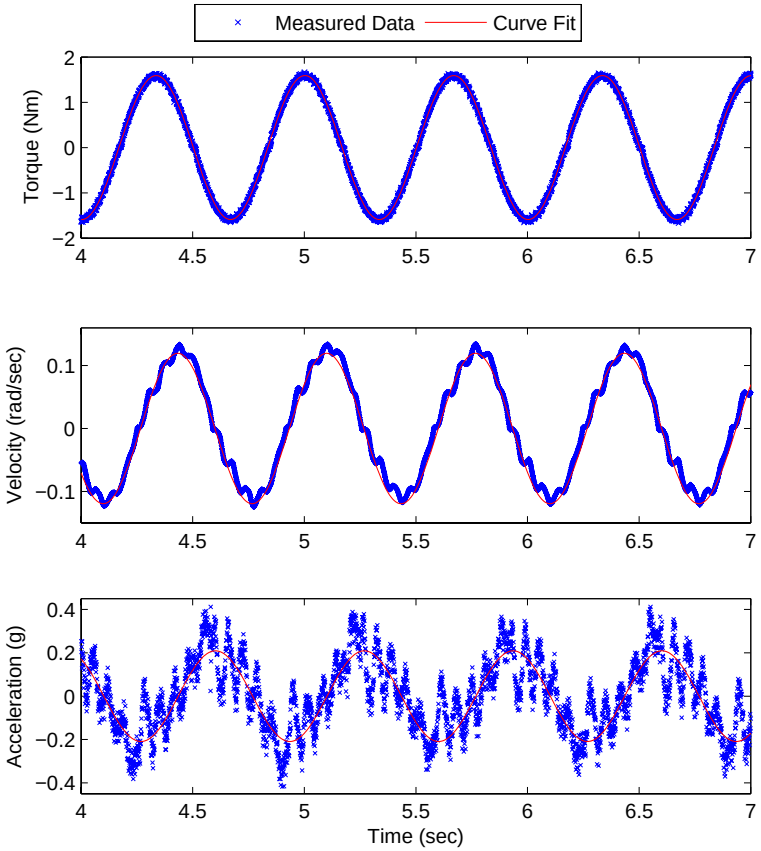


Figure 6.7: Measured data and curve fitting results for J_1

6.3.5 Results

As previously mentioned, the system identification experiments are performed three times, each with different system configurations. For the first set, the joints not being actuated are held rigid by the feedback controllers. Additionally, both open and closed loop experiments are conducted for J_1, J_4, J_5, and J_6. Figure 6.8 shows a comparison between typical closed loop and open loop frequency responses. From the figure, it can be seen that amplitude and phase of the closed loop response closely tracks those of the open loop response. As a result, it can be concluded from Figure 6.8 that using only a closed loop measurement for J_2 and J_3 will not compromise the accuracy of the estimated parameters.

After the initial set of experiments, however, the time history measurements revealed a significant amount of joint coupling during the tests. The feedback controller itself is not enough to hold the other joints at the same position. Since

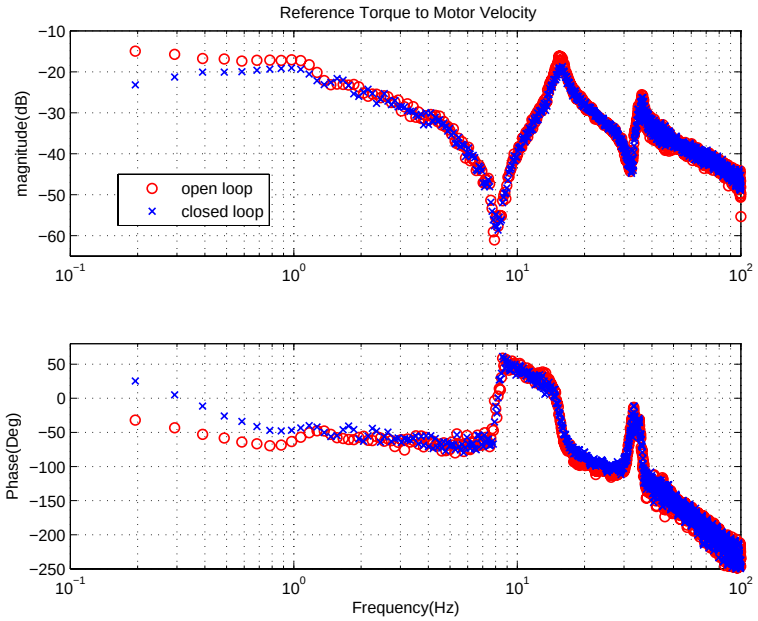


Figure 6.8: Closed-loop and open-loop comparison

the two inertia model used to fit the transfer function assumed the joints to be decoupled, the joint coupling effects could compromise the integrity of the measured data. To address the joint coupling effects, the robot's mechanical brakes were used to lock the unactuated joints in place. When the experiment was redone using the mechanical brakes, the measured time histories showed a significant decrease in the joint coupling effects. The decreased coupling motions, however, may still produce inaccuracies in the parameter estimation. Hence the experiments were performed for a third time, again with the mechanical brakes but without the payload. By removing the payload, the inertial forces responsible for the joint coupling motions will be decreased. This experimental setup produced measurements for J_1, J_2, and J_3 which had almost no joint coupling motions. The lack of a payload, however, did not leave enough system inertia to produce an accurate system identification for J_4, J_5, and J_6. As a result, the presented data in this chapter will take the J_1, J_2, J_3 estimates from the third set of experiments and J_4, J_5, J_6 estimates from the second set of experiments. Additionally, (2.5) is used to fit the parameters. The estimated parameter values are tabulated in Table 6.4.

Table 6.4: Identified system parameters

Parameters	Units	J_1	J_2	J_3
J_ℓ	kgm^2	7.9873×10^1	1.4508×10^2	8.2890
J_m	kgm^2	1.5281×10^{-3}	1.4782×10^{-3}	3.7230×10^{-4}
k_j	N/m	4.5353×10^5	6.9777×10^5	2.4689×10^5
d_j	Ns/m	1.1275×10^3	2.3697×10^3	2.2905×10^2
d_ℓ	Ns/m	≈ 0	1.4481×10^2	6.8287×10^1
d_m	Ns/m	5.3654×10^{-3}	≈ 0	≈ 0
Parameters	Units	J_4	J_5	J_6
J_ℓ	kgm^2	1.0123	3.2690×10^{-1}	3.8485×10^{-1}
J_m	kgm^2	3.1020×10^{-4}	3.9000×10^{-5}	6.4300×10^{-5}
k_j	N/m	7.0821×10^3	2.2765×10^4	1.8283×10^4
d_j	Ns/m	9.7423	7.1963×10^1	5.2327×10^1
d_ℓ	Ns/m	1.1454	≈ 0	≈ 0
d_m	Ns/m	≈ 0	1.2820×10^{-3}	1.9362×10^{-3}

6.4 MIMO System Identification

6.4.1 Problem Formulation

In this section, the problem involving identification of MIMO systems from noise corrupted observations of their responses from n inputs to m outputs is addressed. More specifically, the m outputs and n inputs are related according to

$$y_i = \sum_{j=1}^{n} G_{ij}(q)u_j + \sum_{s=1}^{r} H_{is}(q)e_s \quad \text{for } i = 1, \cdots, m \tag{6.5}$$

where q is the one step advance operator. The scalar transfer function $G_{ij}(q)$ describes the unknown system dynamics that are to be identified. The output measurements y_i are corrupted by zero mean stationary white noise processes e_r with finite variances, and colored by stable filters $H_{is}(q)$. It is possible to simplify (6.5) by using

$$y \triangleq [y_1, y_2, \ldots, y_m)]^T, u \triangleq [u_1, u_2, \ldots, u_n]^T, e \triangleq [e_1, e_2, \ldots, e_r]^T$$

$$G(q) = \begin{bmatrix} G_{11}(q) & G_{12}(q) & \cdots & G_{1n}(q) \\ G_{21}(q) & G_{22}(q) & \cdots & G_{2n}(q) \\ \vdots & \vdots & \cdots & \vdots \\ G_{m1}(q) & G_{m2}(q) & \cdots & G_{mn}(q) \end{bmatrix}$$

$$H(q) = \begin{bmatrix} H_{11}(q) & H_{12}(q) & \cdots & H_{1r}(q) \\ H_{21}(q) & H_{22}(q) & \cdots & H_{21}(q) \\ \vdots & \vdots & \cdots & \vdots \\ H_{m1}(q) & H_{m2}(q) & \cdots & H_{mr}(q) \end{bmatrix}$$

so that (6.5) can be rewritten in matrix form

$$y = G(q)u + H(q)e \tag{6.6}$$

The idea of MIMO system identification is to follow the example of the SISO case and obtain the transfer function matrix $G(q)$ by curve fitting the experimental data.

6.4.2 Identification Approach

The FANUC M-16iB robot consists of six joints. Therefore, from (6.5), ignoring the noise, the input/output relationship can be obtained as

$$y_{mvi} = \sum_{j=1}^{6} G_{ij}(q)u_j \quad \text{for } i = 1, \cdots, 6 \tag{6.7}$$

where u_j is the j^{th} input that's being excited and y_{mvi} is the motor velocity measured from the i^{th} joint. $G_{ij}(q)$ denotes the transfer function that relates the j^{th} input with the i^{th} output.

Figure 6.9 shows the controller structure of the FANUC M-16iB robot. The linear transfer function $G_{ij}(s)$ represents the approximation of robot dynamics. u_{fi} denotes the feedforward torque for the i^{th} joint. Notice that in this chapter, u_{fi} contains only the gravity compensation force and other feedforward torques such as Coulomb and friction compensation terms are not included. Thus, it should be relatively constant for small motions. The control algorithm for the M-16iB robot is implemented in a decentralized form where $C_i(s)$ is the PID controller for the i^{th} joint as explained in Section 6.3.2. d_i is the disturbance torque used to excite i^{th} joint for the system identification purpose.

Figure 6.10 gives one example of the time responses for the identification of $G_{11}(s)$. Supposing that J_1 is excited at a particular frequency by d_1, u_1 is expected to be a combination of d_1, u_{f1}, and $C_1(s)y_{m1}$. The effects of d_1, however, can be isolated if the robot motions are small and $C_1(s)$ is intelligently designed. In particular, if the robot motions are small, the feedforward gravity compensation

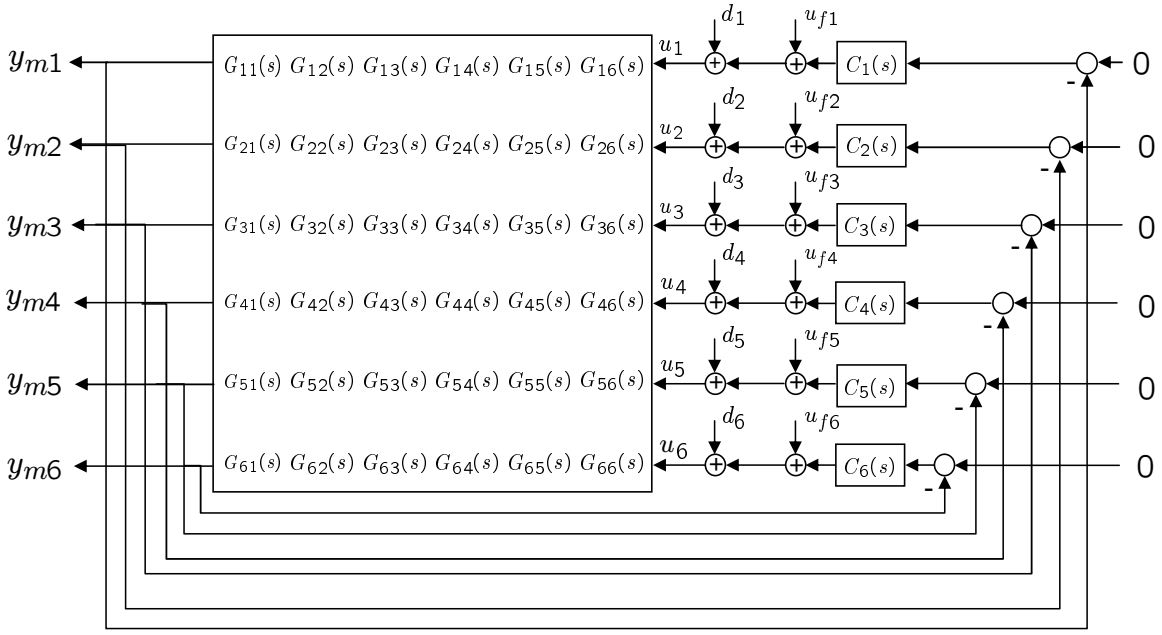


Figure 6.9: Controller structure of M-16iB robot system

is approximately constant and $C_1(s)$ can be designed similarly to the controllers used for the closed loop SISO system identification. As a result, the harmonic component of the input comes solely from d_1. Notice that only J_1 is excited, i.e., d_2,$\cdots$,d_6, are zero. Therefore, u_2,$\cdots$,d_6, only contains the feedforward torque which is approximately constant.

$$\begin{aligned} y_{m1}(t) \quad &= G_{11}(s)u_1(t) + G_{12}(s)u_2(t) + G_{13}(s)u_3(t) \\ &\quad + G_{14}(s)u_4(t) + G_{15}(s)u_5(t) + G_{16}(s)u_6(t) \end{aligned}$$

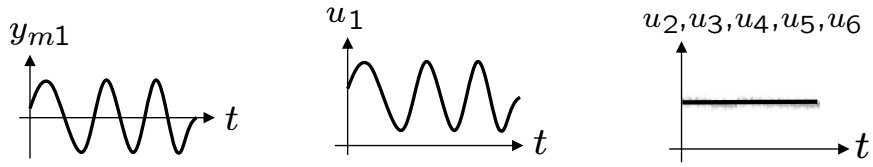


Figure 6.10: Example of the time responses for the identification of $G_{11}(s)$

After obtaining the time domain measurement data, a fast Fourier transform (FFT) can then be used to decompose the inputs, u_i, and the output, y_{mi}, signals into the frequency domain. At this point, the input and output magnitude and

phase at the excitation frequency, ω, can be obtained. Figure 6.11 shows the data processing procedure for $G_{11}(s)$. The input magnitude and phase are defined as $|U_1(\omega)|$ and $\Phi_{U1}(\omega)$ while the output magnitude and phase are defined as $|Y_{m1}(\omega)|$ and $\Phi_{Y_{m1}}(\omega)$, respectively. The amplitude and phase of $G_{11}(\omega)$ can be obtained through amplitude division and phase subtraction of the input output signals. The process shown in Figure 6.11 can be repeated with each input and output, u_j and y_{mi}, to obtain the amplitude and phase for $G_{ij}(\omega)$. And finally, repeating these steps across many frequencies will produce a frequency response for the transfer functions.

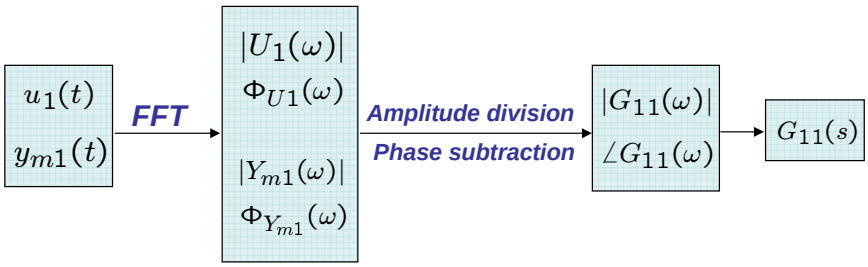


Figure 6.11: MIMO system identification procedure for $G_{11}(s)$

It is always a key problem in system identification to find a suitable model structure and within which a good model is to be found. For the MIMO system identification presented in this chapter, black-box linear models are used to describe/approximate the system's frequency response. The approximation problem has been extensively and successfully handled within some well known linear black-box structures [47, 52]. Once a model structure is chosen (e.g., the order and relative degree of a transfer function), it should be matched as good as possible with the frequency response data obtained from the experiments. Mostly, this is done by minimizing a criterion that measures a goodness of the fit. The procedure is similar to that done in Section 6.3.2.

The robot posture for the MIMO system identification experiments is shown in Figure 6.2. It is known as the zero position where each motor angular position is zero. Different from the SISO system identification, the results obtained through the MIMO system identification approach is valid locally around this posture. In order to obtain a nominal robot model with an uncertainty bound for a desired operational trajectory, the MIMO system identification should be repeated at different postures along the trajectory.

6.4.3 Experimental Procedure

The experimental procedure for MIMO system identification is similar to that of SISO system identification as described in Section 6.3.3. Both the sine sweep and sine by sine tests are done for the robot at the zero position. Opposite from what was explained in the previous section, the coupled joint motions are desired to be identified. Thus, the previously mentioned methods to reduce joint coupling are not used during the MIMO system identification. More specifically, the unactuated joints are held in place through closed loop control while the controller gains are reduced to approximate the open loop system. The payload also is left attached to the robot during the experiments.

6.4.4 Results

The frequency responses from the input torque, u, to the motor velocity of each joint, y_m are obtained. Figure 6.12 and Figure 6.13 show the frequency responses when actuating J_1 and J_2, respectively. It is well known that if one joint, J_p, is decoupled from the actuated joint, J_q, the frequency response, $G_{pq}(s)$, should have really small magnitude values. It can be observed from Figure 6.12 that the frequency responses of $G_{12}(s)$-$G_{16}(s)$ have very small magnitudes relative to the frequency response of $G_{11}(s)$. This is reasonable since the movement of J_1 is not coupled with other joints in the M-16iB robot and therefore the coupling between J_1 and the other joints while J_1 being actuated is very small. Note that for the M-16iB robot, J_2, J_3, and J_5 have the same rotation direction and the coupling effect can be seen from Figure 6.13 where the magnitude plots of $G_{22}(s)$, $G_{23}(s)$, and $G_{25}(s)$ have larger values compared to the frequency responses of G_{21}, G_{24}, and G_{26}.

Other than checking the magnitudes of the frequency responses from the measurement data, the coherence is often used as a measure of the linear dependence of the output on the input. It expresses the degree of linear correlation in the frequency domain between the input and the output signal. The coherence function, $\gamma^2_{xy}(f)$, is given by

$$\gamma^2_{xy}(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)} \tag{6.8}$$

where S_{xx} and S_{yy} are the autospectral densities of the input and output signals, respectively and S_{xy} is the cross-spectral density between the input and the output signals. By definition, the coherence function lies between 0 and 1 for all frequencies f:

$$0 \leq \gamma^2_{xy}(f) \leq 1$$

If $x(t)$ and $y(t)$ are completely unrelated, the coherence function will be zero, while a totally noise-free linear system would yield $\gamma^2_{xy}(f) = 1$. The coherence function may

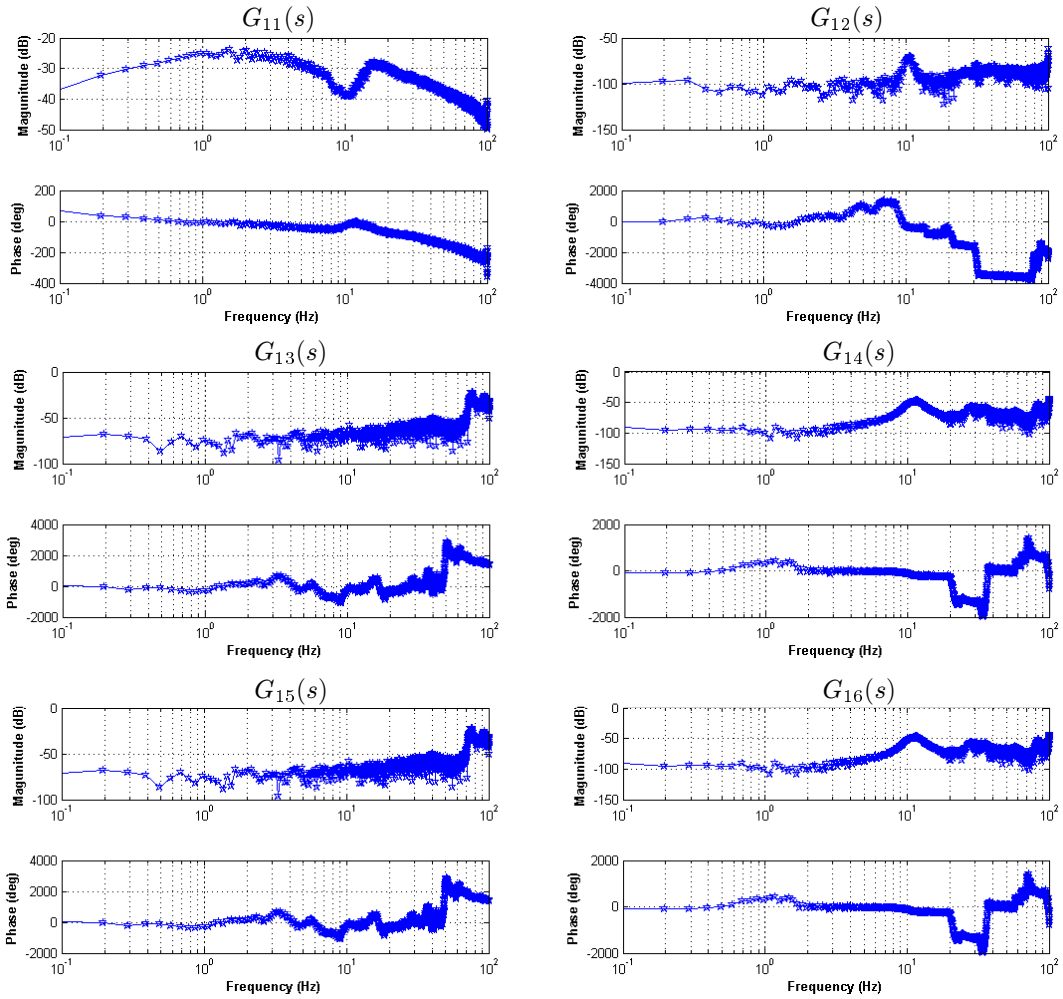


Figure 6.12: Frequency responses of MIMO system identification from u_1

thus be viewed as a type of correlation function in the frequency domain. Note that when a system is noisy or non-linear, the coherence function indicates the accuracy of a linear identification as a function of frequency. A coherence test is therefore employed on all the input-output data channels and is discussed next.

The linearity of the operating region is confirmed by a flat coherence of unity between the input signal and the output responses. The coherence test for all six joints were performed. Figure 6.14 shows the coherence spectra for the nine channels for the first three joints. Good excitation was achieved from 0-100 Hz, which includes all the important rigid-body modes. Strong interaction was observed among the channels u_1 to y_{m1}, u_2 to y_{m1}, u_2 to y_{m2}, u_2 to y_{m3}, u_3 to y_{m1}, u_3 to y_{m2}, and u_3 to y_{m3}. Non-interaction between u_1 to y_{m2}, and u_1 to y_{m3} can also be clearly

visible in the figure. Since no strong coherence exists in the u_1 to y_{m2}, and u_1 to y_{m3} channels, they are not investigated further for model fitting.

As a preliminary study, Figure 6.15 shows the curve fitting result for $G_{11}(s)$. The number of the pole and zero were first chosen from the frequency response plot based on experience. Then, a least squares method as explained in Section 6.3.4 was used to find the transfer function of the frequency response. The transfer function obtained from the curve fitting result was found to be

$$G_{11}(s) = \frac{65.48s^2 + 1834s + 282300}{0.1001s^3 + 6.742s^2 + 1038s} \tag{6.9}$$

Note that the system order for the transfer function, $G_{11}(s)$, was chosen based on experience. The model structure selection will be an interesting future research topic. Furthermore, an uncertainty bound should be identified for the design of a robust linear MIMO controller (e.g. H_∞ robust control design methods). Note that the error bound obtained includes the disturbances, noises and unmodeled dynamics.

6.5 Summary

Chapter 6 presented some system identification work for the FANUC M-16iB robot. The SISO, and MIMO identification experiments were performed and discussed. For SISO system identification, both closed and open loop techniques were used. The experimental procedure was presented and the identified parameter values and frequency characteristics obtained from the experimental results were shown to be reasonable. These parameters should be useful in future research. Next, the approach, procedure, and results for MIMO system identification experiments were presented. There are still a number of problems for MIMO system identification. For example, the transfer functions for the nominal model should be obtained using nonlinear optimization algorithm. The results, however, confirmed the validity of the identification method and should also be useful in future research.

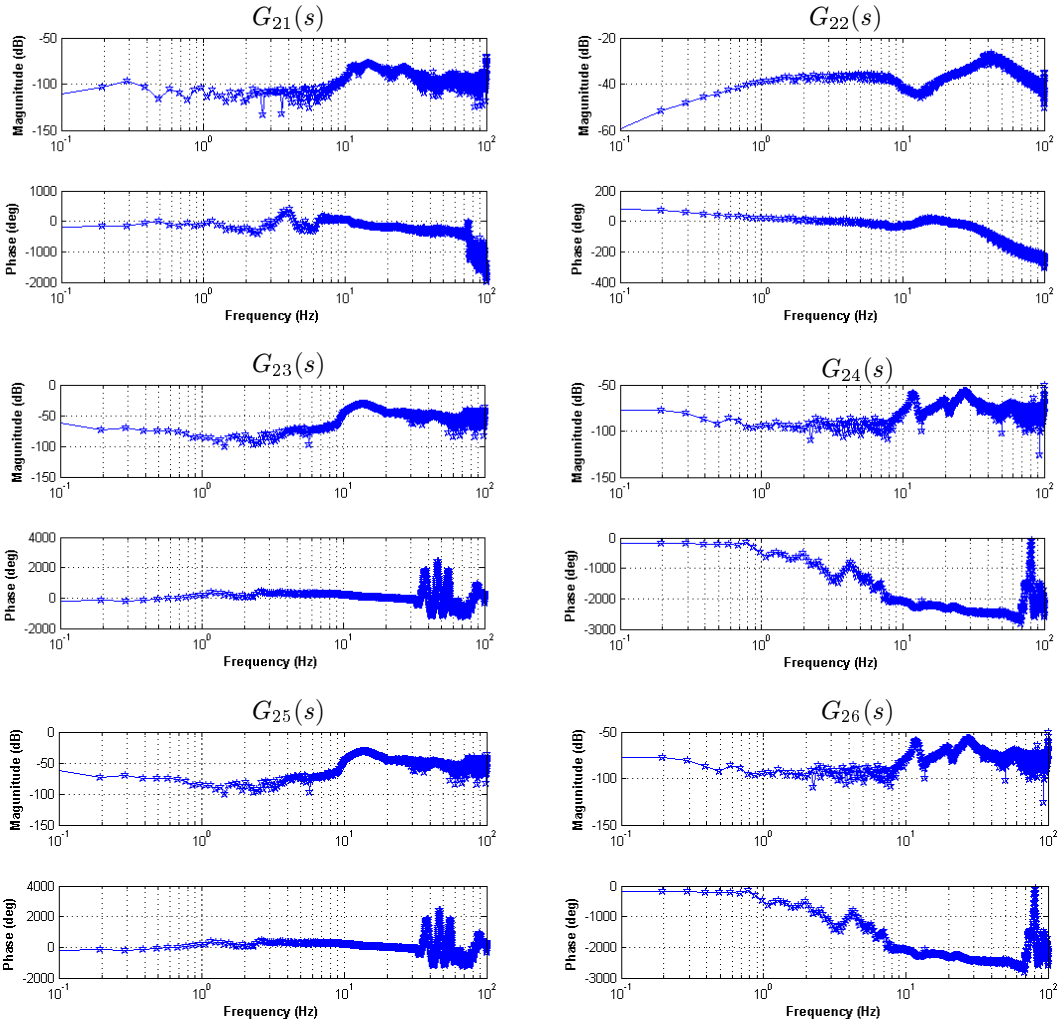


Figure 6.13: Frequency responses of MIMO system identification from u_2

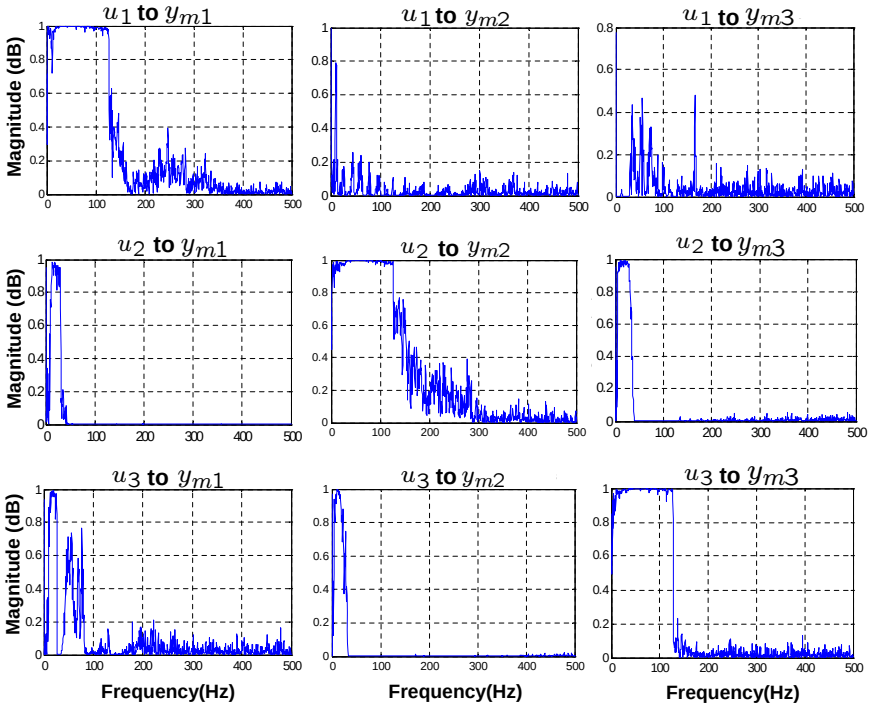


Figure 6.14: Coherence spectrum of MIMO system identification

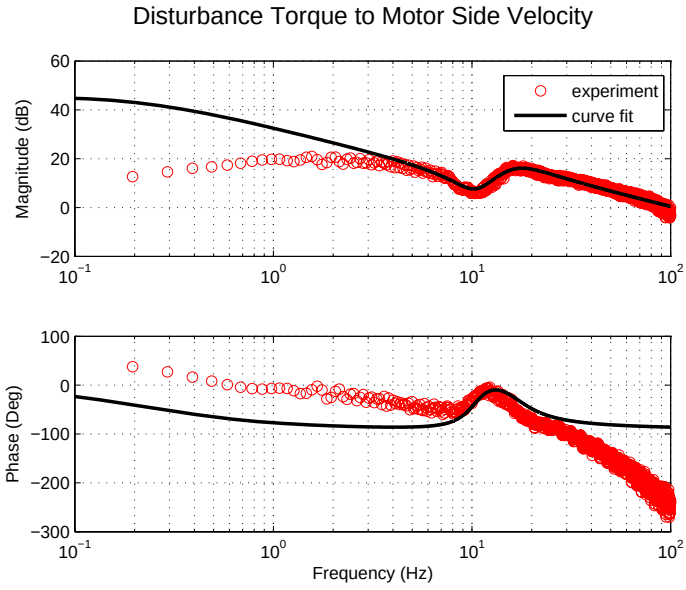


Figure 6.15: Curve fitting result for $G_{11}(s)$

Chapter 7

Conclusion

7.1 Contributions

In the control of indirect drive systems, precise end-effector (load side) tracking, is an important requirement. The nonlinearities and flexibilities associated with speed reducers, such as friction, hysteresis, and transmission error, however, make it difficult to design servo controllers that guarantee good performances. To overcome the limitations in indirect drive train systems, this dissertation suggested new perspectives on the design of servo systems.

In Chapter 3, the controller gains for a single-joint indirect drive train system were optimized by the real-time nonlinear programming (RTNLP) which involved the estimation of the gradient of the cost function by perturbing the magnitude of the controller gains. The RTNLP method tuned the controller gains by minimizing a cost function that characterizes the desired behavior of the closed-loop system. Thus, for the robot system including flexibilities introduced by the harmonic reducer, the load side information could be easily included in the cost function to reflect the desired performance attributes. For this purpose, the acceleration measurement obtained from accelerometers mounted on the payload of the indirect drive train was used as an indicator of the load side performance for gain tuning because of easy installation of MEMS accelerometers. Eleven controller parameters including the feedforward, the feedback, and the state feedback controller gains were tuned. The experimental results were compared with the iterative feedback tuning method. The effectiveness of the RTNLP method was demonstrated by experiments and the tuned controller gains were shown to be suitable for achieving better load side performance. Moreover, unlike the iterative feedback tuning method, the RTNLP method did not require any additional experiments for the estimation of gradient and the optimization process was totally automated.

Next, an adaptive cancellation algorithm to suppress the oscillatory phenomena caused by the transmission error in harmonic drives was presented in Chapter 4.

The transmission error was introduced and modeled. Then, an analysis scheme that considers the transmission error effect as a disturbance input to the system was constructed. The adaptive disturbance cancellation algorithm was derived where the acceleration information was used in the adaptation laws to effectively reduce the oscillations on the load side. To enhance the disturbance rejection performance while maintaining the transient responses, two modifications were proposed. The variable step-size scheme adaptively changed the update gains to avoid deteriorating the transient response. The compensation scheme, which considers the measurement bias, further improved the performance at the steady state by providing more accurate load side acceleration information. Experimental results showed the effectiveness of the proposed methods, and the load side oscillation caused by the transmission error was significantly reduced.

In Chapter 5, an optimization-based iterative learning control scheme for the purpose of disturbance rejection was proposed. The approach is a systematic and model-based design method, which allows to handle the trade-off between performance and robustness. Two different learning controllers, the motor side and load side learning controllers, designed based on different error signals, namely the motor side error and the load side errors, were obtained. Even though the experimental setup used was equipped with a load side encoder, the load side position information was only used for the performance evaluation purpose. Therefore, a load side position estimation algorithm based on Kalman filtering using load side acceleration measurement was proposed, and the load side position estimate was used in the load side learning controller. The effectiveness of the estimation scheme was shown by experiments. The load side iterative learning controller demonstrated more substantial reduction of the load side vibration compared to the motor side learning controller.

Finally, Chapter 6 presented the system identification of the FANUC M-16iB robot. The SISO and MIMO identification experiments were performed and discussed. For SISO system identification, both closed and open loop techniques were used. The experimental procedure was presented and the identified parameter values and frequency characteristics obtained from the experimental results were shown to be reasonable. The results will be useful in future research. Next, the approach, procedure, and results for MIMO system identification experiments were presented. The MIMO system identification results should be viewed as preliminary results and there are still a number of problems to deal with. The results, however, confirmed the validity of the proposed MIMO identification approach and should also be useful in future research.

7.2 Future Research

The results in this dissertation raise several issues and can be expanded as future research topics.

- Extensive investigation on the RTNLP method: In Chapter 3, the feedback controllers were tuned based on a filtered step motor reference trajectory. Even though in most industrial applications, the desired trajectories are normally known in advance and the controller gains are usually tuned for the specified trajectory, it is desirable to investigate the influences of different reference trajectories on the resulting optimal controller gains. Moreover, the investigation of different performance indices for optimizing the controller gains and their effects on trajectory tracking performance may be worth investigating. Furthermore, the tuning method should be extended and implemented for the tuning of controllers that consider nonlinearities such as input/output saturation and friction.

- Generalization of the SISO control algorithms to MIMO settings that will include all six robot axes simultaneously: The control algorithms developed in Chapters $3 \sim 5$ were successfully implemented on the single-joint indirect drive train setup. The algorithms need to be tested in MIMO settings for applications to robot manipulators and other industrial motion control systems. The FANUC M-16iB available in the Mechanical Systems and Control Laboratory at Berkeley is an excellent testbed for this purpose. The SISO method should be first applied on the first joint of the M-16iB robot. Then, the algorithms for multi-joint robots can be developed with a decentralized scheme by applying the SISO method to each joint.

- MIMO system identification of robot dynamics: The MIMO system identification results were viewed as preliminary results and there are still a number of problems to deal with. The order of the transfer function for fitting purposes in Chapter 6 was chosen based on experience. The model structure selection and the nonlinear optimization algorithm for transfer function fitting should be explored. Moreover, uncertainty bounds for disturbances and unmodeled dynamics should be estimated for the design of a robust linear MIMO controller (e.g. H_∞ robust control design) Furthermore, the nominal model should be obtained for each of different configurations of the robot.

Bibliography

[1] 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.

[2] J. S. Arora, *Introduction to Optimum Design*, 2nd ed., Academic Press, 2004.

[3] N. Bajçinca, R. Cortes ao, and M. Hauschild, "Robust control for steer-by-wire vehicles," *Autonomous Robots*, vol. 19, no. 2, pp. 193–214, 2005.

[4] Z. Bien and J. Xu, Eds., *Iterative learning control: analysis, design, integration and applications*, Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[5] M. Bodson, A. Sacks, and P. Khosla, "Harmonic generation in adaptive feedforward cancellation schemes," *IEEE Transactions on Automatic Control*, vol. 39, no. 9, pp. 1939–1944, 1994.

[6] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, June 2006.

[7] B. Bukkems, D. Kostic, B. de Jager, and M. Steinbuch, "Learning-based identification and iterative learning control of direct-drive robots," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 537–549, July 2005.

[8] C. Canudas de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 419–425, 1995.

[9] W. Chen, K. Kong, and M. Tomizuka, "Hybrid adaptive friction compensation of indirect drive trains," *Proceedings of the ASME International Conference on Dynamic Systems and Control*, 2009.

[10] W. Chen and M. Tomizuka, "Estimation of load side position in indirect drive robots by sensor fusion and kalman filtering," *Submitted to the 2010 American Control Conference*, 2010.

[11] Y. Chen, C. Wen, J. Xu, and M. Sun, "An initial state learning method for iterative learning control of uncertain time-varying systems," *Proceedings of the 35th IEEE conference on Decision and Control*, vol. 4, Dec 1996, pp. 3996–4001.

[12] C. Chien, "A discrete iterative learning control of nonlinear time-varying systems," *Proceedings of the 35th IEEE conference on Decision and Control*, vol. 3, Dec 1996, pp. 3056–3061.

[13] Dynalog, Inc., *CompuGauge*, http://www.dynalog-us.com/solutions.

[14] R. A. de Callafon, "Feedback oriented identification for enhanced and robust control - a fractional apporach applied to a wafer stage," *Ph.D. Dissertation*, Technical University of Delft, The Netherlands, 1998.

[15] J. Doyle, M. Newlin, F. Paganini, and J. Tierno, "Unifying robustness analysis and system id," *Proceedings of the 33rd IEEE Conference on Decision and Control*, vol. 4, Dec 1994, pp. 3667–3672.

[16] ENCYCLOPEDIA Britannica, http://www.britannica.com/, *Robot Manipulator*.

[17] J. Frueh and M. Phan, "Linear quadratic optimal learning control (LQL)," *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 1, 1998, pp. 678–683.

[18] P. S. Gandhi and F. H. Ghorbel, "Closed-loop compensation of kinematic error in harmonic drives for precision control applications," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 6, pp. 759–768, 2002.

[19] F. H. Ghorbel, P. S. Gandhi, and F. Alpeter, "On the kinematic error in harmonic drive gears," *Journal of Mechanical Design*, vol. 123, no. 1, pp. 90–97, 2001.

[20] I. Godler, K. Ohnishi, and T. Yamashita, "Repetitive control to reduce speed ripple caused by strain wave gearing," *Proceedings of the 20th International Conference on Industrial Electronics, Control and Instrumentation*, vol. 2, 1994, pp. 1034–1038.

[21] D. Gorinevsky, D. Torfs, and A. Goldenberg, "Learning approximation of feedforward dependence on the task parameters: Experiments in direct-drive manipulator tracking," *Proceedings of the American Control Conference*, vol. 1, Jun 1995, pp. 883–887.

[22] A. Graham, A. Young, and S. Xie, "Rapid tuning of controllers by ift for profile cutting machines," *Mechatronics*, vol. 17, no. 2-3, pp. 121 – 128, 2007.

[23] S. Gunnarsson and M. Norrlof, "On the design of ilc algorithms using optimization," *Automatica*, vol. 37, no. 12, pp. 2011 – 2016, 2001.

[24] K. Hamamoto, T. Fukuda, and T. Sugie, "Iterative feedback tuning of controllers for a two-mass spring system with friction," *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 3, 2000, pp. 2438–2443.

[25] C. Han, W. Chen, P. R. Mora, M. Chan, and M. Tomizuka, *UCB-FANUC Project Activity Report*, University of California, Berkeley, May 2009.

[26] Harmonic Drive Technologies, "HDC cup component gear set selection guide." http://www.HarmonicDrive.net/

[27] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice Hall, 2001.

[28] H. Hirabayashi, J. Chiba, and A. Akahane, "Vibration suppression of strain wave gearing," *Proceedings of the JSME Conference on Robotics and Mechatronics, Osaka, Japan, Jun 8-9*, vol. 2, 1990, pp. 1034–1038.

[29] G. Hirzinger, N. Sporer, M. Schedl, J. Butterfass, and M. Grebenstein, "Torque-Controlled Lightweight Arms and Articulated Hands: Do We Reach Technological Limits Now?" *International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 331–340, 2004.

[30] H. Hjalmarsson, "Iterative feedback tuning - an overview," *International Journal of Adaptive Control and Signal Processing*, vol. 16, no. 5, pp. 373–395, 2002.

[31] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: theory and applications," *IEEE Control Systems Magazine*, vol. 18, no. 4, pp. 26–41, Aug 1998.

[32] H. Hjalmarsson, S. Gunnarsson, and M. Gevers, "A convergent iterative restricted complexity control design scheme," *Proceedings of the 33rd IEEE Conference on Decision and Control*, vol. 2, Dec 1994, pp. 1735–1740.

[33] R. Horowitz, "Learning control of robot manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, no. 115, pp. 403–411, 1993.

[34] J. Huusom, H. Hjalmarsson, N. Poulsen, and S. Jorgensen, "Improving convergence of iterative feedback tuning using optimal external perturbations," *Proceedings of the 47th IEEE Conference on Decision and Control*, Dec. 2008, pp. 2618–2623.

[35] J.H. Lee, K.S. Lee, W.C. Kim",, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, pp. 641–657, May 2000.

[36] S. Kawamura, F. Miyazaki, and S. Arimoto, "Realization of robot motion based on a learning method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 126–134, 1988.

[37] N. Killingsworth and M. Krstic, "Pid tuning using extremum seeking: online, model-free performance optimization," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 70–79, Feb. 2006.

[38] Kistler, *The accelerometer*, http://www.kislter.com/mediaaccess.

[39] S. Komada, K. Iyama, K. Yubai, and T. Hori, "Suppression of limit cycle and improvement of robust performance in two-mass resonant systems with nonlinearity," *Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics Society*, vol. 3, 2001, pp. 1704–1709.

[40] K. Kong, K. Inaba, and M. Tomizuka, "Real-time nonlinear programming by amplitude modulation," *Proceedings of the ASME International Conference on Dynamic Systems and Control*, 2008.

[41] R. H. Kwong and E. W. Johnston, "A variable step size lms algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1633–1642, 1992.

[42] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Engineering Practice*, vol. 13, no. 2, pp. 145 – 157, 2005.

[43] F. Lange and G. Hirzinger, "Learning accurate path control of industrial robots with joint elasticity," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, 1999, pp. 2084–2089.

[44] MATLAB, http://www.mathworks.com/products/index.html.

[45] K. L. Moore, *Iterative Learning Control for Deterministic Systems*, Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1993.

[46] National Instrument, *LabVIEW*, http://www.ni.com/labview/.

[47] B. Ninness, J. C. Gomez, and S. Weller, "MIMO system identification using orthonormal basis functions," *Proceedings of the 34th IEEE Conference on Decision and Control*, vol. 1, Dec 1995, pp. 703–708.

[48] M. Norrlof, "Disturbance rejection using an ilc algorithm with iteration varying filters," *Asian Journal of Control*, vol. 6, no. 3, pp. 432–438, 1998.

[49] 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, Aug 2002.

[50] T. W. Nye and R. P. Kraml, "Harmonic Drive Gear Error: Characterization and Compensation for Precision Pointing and Tracking," *Proceedings of the 25th Aerospace Mechanisms Symposium*, ser. NASA Conference Publication, vol. 3113, 1991.

[51] K. Ohnishi, "A new servo method in mechatronics," *Transaction of Japanese Society of Electrical Engineering*, vol. 107-D, pp. 83–86, 1987.

[52] A. Rahideh, M. Shaheed, and H. Huijberts, "Dynamic modelling of a trms using analytical and empirical approaches," *Control Engineering Practice*, vol. 16, no. 3, pp. 241 – 259, 2008.

[53] D. D. Roover and O. H. Bosgra, "Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system," *International Journal of Control*, vol. 73, pp. 968–979(12), 10 July 2000.

[54] A. H. Sacks, M. Bodson, and W. Messner, "Advanced methods for repeatable runout compensation [disc drives]," *IEEE Transactions on Magnetics*, vol. 31, no. 2, pp. 1031–1036, 1995.

[55] M. Saeki, "Unfalsified control approach to parameter space design of pid controllers," *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 1, Dec. 2003, pp. 786–791.

[56] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, Wiley-Interscience, 2006.

[57] H. D. Taghirad and P. R. Belanger, "Modeling and parameter identification of harmonic drive systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 120, no. 4, pp. 439–444, 1998.

[58] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 1, pp. 65–68, 1987.

[59] M. Tomizuka, "Mechatronics: from the 20th to 21st century," *Control Engineering Practice*, vol. 10, no. 8, pp. 877 – 886, 2002.

[60] T. D. Tuttle, "Understanding and Modeling the Behavior of a Harmonic Drive Gear Transmission," *NASA STI/Recon Technical Report N*, vol. 93, May 1992.

[61] T. D. Tuttle and W. Seering, "Modeling a harmonic drive gear transmission," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, 1993, pp. 624–629.

[62] M. Ueberle and M. Buss, "Design, control, and evaluation of a new 6 DOF haptic device," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2002, pp. 2949–2954.

[63] T. Umeno and Y. Hori, "Robust speed control of dc servomotors using modern two degrees-of-freedom controller design," *IEEE Transactions on Industrial Electronics*, vol. 38, no. 5, pp. 363–368, Oct 1991.

[64] W. J. R. Velthuis, T. J. A. de Vries, and J. Van Amerongen, "Learning feedforward control of a flexible beam," *Proceedings of the 1996 IEEE International Symposium on Intelligent Control*, 1996, pp. 103–108.

[65] M. C. Walton, "Strain wave gearing", Patent 2 906 143, September, 1959, http://www.freepatentsonline.com/2906143.html.

[66] C. Wang, S. Joen, C. Han, and M. Tomizuka, *UCB-FANUC Project Activity Report*. University of California, Berkeley, May 2007.

[67] C. Wang and M. Tomizuka, "Sensor-based controller tuning of indirect drive trains," *Proceedings of the 10th IEEE International Workshop on Advanced Motion Control*, pp. 188–193, March 2008.

[68] C. Wang, "Motion control of indirect-drive robots: Model based controller design and performance enhancement based on load-side sensors," *Ph.D. Dissertation*, University of California, Berkeley, 2008.

[69] S. Wolfgang and J. Angeles, "A mechanical model for robotic joints with harmonic drives," *Technical Report*, McGill University Department of Mechanical Engineering and Centre for Intelligent Machines, Montreal, Quebec, Canada, 1995.