國立交通大學

電機與控制工程研究所

博士論文

人機系統之虛擬運動限制設計：
模擬系統與機器人操作之應用

Virtual Motion Constraint Design for Human-Machine Systems: Applications in Simulation Systems and Robot Manipulation

研 究 生：謝木政

指導教授：楊谷洋 博士

中華民國一百零一年四月

人機系統之虛擬運動限制設計：模擬系統與機器人操作之應用

Virtual Motion Constraint Design for Human-Machine Systems: Applications in Simulation Systems and Robot Manipulation

研究生：謝木政 Student: Mu-Cheng Hsieh

指導教授：楊谷洋 博士 Advisor: Dr. Kuu-young Young

國立交通大學

電機與控制工程研究所

博士論文

A Dissertation

Submitted to Institute of Electrical Control Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Control Engineering

April 2012

Hsinchu, Taiwan

中華民國一百零一年四月

人機系統之虛擬運動限制設計：
模擬系統與機器人操作之應用

學生：謝木政　　　　　　　　　　指導教授：楊谷洋 博士

國立交通大學電機與控制工程研究所博士班

中文摘要

現今各種觸覺裝置已廣泛應用在娛樂、教育、訓練、製造、機器人遙控操作、藝術等各領域中，藉由觸覺以增加虛擬物件的眞實感或提供力輔助來促進操作感受及提昇工作效能。虛擬運動限制(virtual motion constraint)係由軟體產生，透過力回饋搖桿來限制操作者的運動空間，藉以符合對應的工作要求或提供操作者即時的操作輔助。基於此觀念，本論文提出虛擬運動限制的設計方法，可適用於已知或未定的操作環境，並應用在模擬操控系統及機器手臂操作上。前者，針對已知的模擬工作，我們經由組合虛擬牆來建構運動限制，以兩軸力回饋搖桿發展出一個多功能的虛擬操作系統，可用來模擬多種不同工作的操作裝置，如扳手、排擋器等。我們也提出以像素(pixel)爲基礎的方法，不僅便於以圖形編輯軟體來組裝各式虛擬牆，且能在不同牆面間達成力的平滑呈現。實驗結果顯示，該系統能掌握所模擬操作裝置的主要特徵。而後者，關於未定環境的多軸機器手臂操作，我們以運動限制發展一套虛擬工具，如虛擬尺等，根據使用者的需要，提供使用者在位置與方向上的輔助，經由操作六軸力回饋搖桿來操控機器手臂。我們也提出點、線、面等虛擬工具的實現方法，且能在不同工具或方向切換時，仍使輔助力保持平順。爲了展現成果，在實驗中本系統執行路徑跟隨以及用扳手鎖螺絲這類需要技巧的操作工作。

Virtual Motion Constraint Design for Human-Machine Systems: Applications in Simulation Systems and Robot Manipulation

Student: Mu-Cheng Hsieh Advisor: Dr. Kuu-young Young

Institute of Electrical Control Engineering
College of Electrical and Computer Engineering
National Chiao Tung University

Abstract

Nowadays haptic devices have been applied to many areas, such as entertainment, education, training, manufacturing, telerobotics, art, and so on, to enhance the reality of virtual objects or provide force assistance, and thus to foster manipulative feeling or to improve task performance. Virtual motion constraint is generated via the software, so that the force-reflection joystick, operated by the user, is confined to move within a limited workspace that corresponds to task requirements or provides the user the real-time assistance. Based on this concept, this dissertation proposes methods to design virtual motion constraints, which can be applicable for well-known or uncertain environments, and apply them for simulation systems and robot manipulation. First, for a well-known simulated task, we assembly virtual walls to construct the desired motion constraint and thus develop a multi-functional virtual manipulation system, based on a 2-DOF (degree-of-freedom) force-reflection joystick, to emulate various manipulative devices, e.g., a wrench or gearshift level. We also propose a pixel-based method, which can easily construct these motion constraints by using the graphic editing software and also maintain smooth force rendering between the walls. The experimental results show that this system can capture the main features of various kinds of manipulative devices. Second, for manipulating a multi-DOF robot manipulator in uncertain environments, also based on the concept of virtual motion constraint, we develop a set of virtual tools, e.g., a virtual ruler, which can provide the position or orientation assistance to the user, according to her/his demand on site, to govern the robot effectively by operating a 6-DOF force-reflection joystick. We also implement several kinds of virtual tools, such as a point, line, or plane. Smooth force rendering during the transition between the use of two consecutive tools in guidance are also achieved. For demonstration, the system is employed for the tasks of contour following and screw fastening, which involve delicate and challenging maneuvers.

Acknowledgment

首先我要衷心感謝我的指導老師楊谷洋教授，感謝他的體諒關照，我才能兼顧工作來攻讀博士學位，當我困惑氣餒時，他的熱情指導與鼓勵，總能帶給我希望以及繼續向前的動力，今日我能完成夢想取得博士學位，全歸功於老師的耐心教導。

感謝柯春旭老師、邱添丁老師、林顯易老師、陳永平老師、周志成老師、與林文杰老師，在口試時給予許多的寶貴指導與建議，讓我受益良多，謹此對所有老師致上最大謝意。

謝謝這些年來曾經在「人與機器實驗室」與我共同學習的所有同學，這是最友善的實驗室。特別感謝修任、豪宇，在研究上的討論與鼓勵；感謝與我進行機器人研究的合作伙伴—明勳和柏穎，有他們才能完成系統的建置以及實驗工作；也要感謝一哲、嘉文、禹均大力幫忙我順利完成口試的工作。

感謝我工作的清華大學計算機與通訊中心，在我修業期間歷任長官同仁們對我的關心與支持，尤其是唐傳義主任與王家祥主任；我要特別感謝曾共事數年的好友賴守全博士對我的關照；還有感謝同事怡碩這些年來幫我分擔不少的工作。

最後，感謝我的父母以及親愛的家人，特別衷心感謝我太太君幸的支持，這些年來全心全意照顧兒女—易佑與易晴，讓我可以專心學業。要感謝的人實在太多了，無法一一道出，總之，謝謝大家。

Table of Contents

中文摘要 i

Abstract ii

Acknowledgment iv

Table of Contents v

List of Tables vii

List of Figures viii

1 Introduction 1

1.1 Research Motivation 1

1.2 Dissertation Contributions 5

1.3 Dissertation Organization 7

2 Multi-Functional Virtual Manipulation System 8

2.1 Introduction 8

2.2 Proposed Virtual Manipulation System 10

2.2.1 Virtual Wall Design 12

2.2.2 Motion Constraint Construction 14

2.3 Experiments 18

2.3.1 Virtual Omni-Directional Wrench 20

2.3.2 Virtual Manual Gearshift System 25

2.4 User Response Evaluation 27

2.5 Summary 30

3 Effective Manipulation for a Multi-DOF Robot Manipulator 31

3.1 Introduction 31

3.2 Proposed Manipulation System 33

3.2.1 Virtual Linkage 35

3.2.2 Virtual Tools 38

3.2.3 Virtual Bumper 43

3.3 System Implementation 45

3.4 Experiments 47

3.4.1 Executing Linear Movement with Fixed Orientation 48

3.4.2 Manipulating the Hex Key for Screw Fastening 51

3.5 Summary 57

4 Conclusion and Future Work 58

Bibliography 60

Vita 69

List of Tables

2.1 Specifications of the developed 2-DOF force-reflection joystick. 20

3.1 Joint specifications of Mitsubishi RV-2A robot arm 47

3.2 Number of failures for the four operation modes 53

List of Figures

2.1 Conceptual diagram of the proposed multi-functional virtual manipulation system. 11

2.2 Virtual aisle design: (a) rectangular and circular aisles and (b) physical property emulation. 13

2.3 Connection between walls: (a) line-to-line, (b) curve-to-curve, and (c) line-to-curve. 15

2.4 Applying the proposed pixel-based method for force rendering for wall connection in Fig. 2.3(a). 16

2.5 (a) System view of the 2-DOF force-reflection joystick system developed in our laboratory. (b) System block diagram with the operator interacting with the simulated environment using the force-reflection joystick. 19

2.6 Motion constraint construction for the virtual omni-directional wrench: (a) manipulation of the real omni-directional wrench, (b) virtual motion constraint generation, and (c) manipulation of the virtual omni-directional wrench. 21

2.7 Measured force response for screw tightening using a real wrench. 22

2.8 Experimental results for applying the virtual omni-directional wrench for screw tightening ($c = 5$ N·s/m·rad): (a) the screw-tightening process, (b) the trajectory of the wrench (in angle), and (c) the measured force response. 24

2.9 Experimental results for applying the virtual omni-directional wrench for both screw tightening and loosening (c = 15 N·s/m·rad): (a) the screw-tightening (clockwise) and loosening (counterclockwise) processes, (b) the trajectory of the wrench (in angle), and (c) the measured force response. . 25

2.10 Experimental results for the emulation of a virtual manual 5-speed gearshift system: (a) the virtual manual 5-speed gearshift system, (b) the position trajectory of the virtual gearshift lever, (c) the position responses along the X and Y axes, and (c) the measured force response. 26

2.11 Evaluation of the users' responses: (a) the omni-directional wrench and (b) the manual gearshift system. 29

3.1 Proposed manipulation system for a position-controlled robot manipulator: (a) conceptual diagram and (b) system block diagram. 34

3.2 The user utilizes the manipulation system to manipulate (a) the real robot manipulator via (b) the virtual robot manipulator. 36

3.3 Alignment and workspace mapping between the coordinate systems of the joystick and robot manipulator. 37

3.4 Proposed virtual tools and their possible applications: (a) point, (b) line, (c) plane, (d) N-rotation, and (e) F-rotation. 38

3.5 The process of how the line tool executes motion assistance. 39

3.6 Two force rendering methods for the transition of two consecutive tools: (a) the orthogonal-projection method (unsmooth force rendering) and (b) the proposed method (smooth force rendering). 40

3.7 Illustration for applying the F-rotation tool for assistance. 42

3.8 System implementation block diagram of the proposed manipulation system. 45

3.9 Experimental setup for executing linear movement from point A to B, then C with fixed orientations. 49

3.10 Box plots for the linear movement task: (a) execution time, (b) robot traveling path, and (c)-(d) maximum position and orientation errors. . . . 50

3.11 Desired and actual robot trajectories for user A in (a) mode NV and (b) mode V. 51

3.12 Experimental setup for manipulating the hex key for screw fastening: (a) place the ball end of the hex key into the hex socket screw and turn it clockwise to fasten the screw and (b) release the hex key from the screw by turning it counter-clockwise. 52

3.13 Box plots for the screw-fastening task: (a) completion time, (b) traveling path, and (c) average contact force. 54

3.14 Robot trajectories for user D in (a) mode NC-NV, (b) mode NC-V, (c) mode C-NV, and (d) mode C-V. 55

3.15 Trajectories of contact force and spring force for user D (Z-axis) in (a) mode NC-NV vs. C-NV and (b) mode NC-V vs. C-V. 56

Chapter 1 Introduction

1.1 Research Motivation

Along with the rapid development of virtual reality (VR) technology and human-machine interface devices in recent years, haptic devices have been applied to many areas, such as entertainment, education, training, industry, manufacturing, telerobotics, art, and so on, to enhance the reality of virtual objects or provide force assistance, and thus to foster manipulative feeling or to improve task performance [24,57]. In human-machine systems, VR is often used to yield the user an illusion of reality and a feeling of "being there" via real-time interaction with her/him through many different sensorial channels, such as vision, audition, and haptics [9–11]. Unlike that a single visual or auditory device can only serve one-way communication with the user, e.g., a display or speaker, a haptic device is bilateral for the user, such as force-feedback gloves or joystick, which enables the user can percept or manipulate the virtual or remote object via physical contact, including both tactile and kinesthetic senses [33,52,56,57].

According to our survey on previous works related to haptic applications in manipulative tasks, basically, there are two major purposes of applying the haptic devices in human-machine systems: i) to emulate real objects or motions, such as telepresence, simulation systems for skill training, haptic display of virtual object, and ii) to provide assistance to the user by virtual objects for guidance or protection. To render the haptic effects, in the former the force information may be obtained from remote force sensors or simulated models, which may include geometric and physical properties, e.g., shape, hardness, or stickiness. We classify them into two categories below:

1. Sensor-based method: force-reflecting teleoperation [23,32]

2. Model-based method:

    - Dedicated-function: simulation systems of calligraphy [36,61] and manual gearshift [18], skill training system for arc welding tasks [62]

    - General-function: 3D haptic display [54,64], *multi-functional virtual manipulation system for simulated tasks* [26]

However, in the latter, the force information comes from the designer, which may be based on predefined environments, predictable user motion or environment, real-time sensed results, or user's choice. We further classify them into four types below:

1. Predefined type: virtual fixtures (VF) [53] and virtual mechanism [28] for telerobotic manipulation, geometrically-based VF in VR training environments [31,49, 50], anatomy-based constraints for surgery tasks [34], velocity-based guidance VF for telemanipulation [4,5]

2. Predicted type:

    - User motion: HMM-based VF for path following and off-path motions [35], adaptive VF for teleoperation tasks [3]

    - Environment: predictive haptic guidance for driving support [17], predicted-position VF for protection in robot-assisted surgery [20]

3. Sensor-based type: vision-based VF for microscale manipulation [8], laser-assisted for telerobotic system [60] or robotic surgery [51]

4. User-based type: *virtual tools for robot manipulation* [25]

According to the survey above, there are two important concepts, which are often used to enhance human performance in manipulative tasks: virtual fixtures and virtual mechanism. In 1993, Rosenberg [53] first introduced the concept of virtual fixtures, which is an overlay of abstract perceptual information on a workspace to improve the human performance in a telerobotic task. For instance, a ruler can assist the user draw a line straighter and faster than freehand. Note that virtual fixtures may include not only haptic information but also visual or audio information. For the applications, virtual fixtures are often used to either serve for guidance, which lets the manipulated object move along desired paths, or to bound forbidden-regions, which protect it from hitting obstacles [4]. In 1995, Joly and Andriot [28] proposed the concept of virtual mechanism also for a telerobotic system. Virtual mechanism can confine both the master and slave arms individually, to move along a desired task space (e.g., a line or a circle), via virtual springs and dampers attached between them, in which the constrained forces are calculated by real-time simulation of this mechanical system.

Motivated by the concepts of virtual fixtures and virtual mechanism, in this dissertation, the virtual motion constraint is generated via the software, so that the force-reflection joystick, operated by the user, is confined to move within a limited workspace that corresponds to task requirements or provides the user the real-time assistance. Therefore, the main purpose of this dissertation is to propose methods to design virtual motion constraints, which can be applicable for well-known or uncertain environments, and apply them for simulation systems and robot manipulation.

First, for a well-known simulated task, we assembly virtual walls to construct the desired motion constraint and thus develop a multi-functional virtual manipulation system, based on a 2-DOF (degree-of-freedom) force-reflection joystick, which can emulate various manipulative devices, e.g., a wrench or gearshift level. Differing from the previous works of dedicated simulation systems [18,36,61,62], which only can emulate a single device for exact replication based on detailed analysis or dedicated hardware, we intend to let this virtual manipulative system be multi-functional via a systematic way to assemble different kinds of virtual walls, which can capture the main features of various kinds of manipulative devices, at the expense of exact replication. We also propose a pixel-based method, which can easily construct these motion constraints by using the graphic editing software and also maintain smooth force rendering between the walls. The experimental results show that this system can exhibit certain degree of resemblance and reality.

Second, for manipulating a multi-DOF robot manipulator in uncertain environments, also based on the concept of virtual motion constraint, we develop a set of virtual tools, e.g., a virtual ruler. These tools can provide the position or orientation assistance to the user, according to her/his call and setup on site, to govern the robot effectively. For instance, in a screw fastening task, the operator first manipulates the hex key, which is mounted on the robot manipulator, into a hex socket screw with a proper orientation. Then, she/he calls the virtual tools to let this hex key maintain a fixed rotation axis and also move only along with this axis. Thus, with the help of virtual tools, the task becomes more easier than the case without them. Differing from the predicted or sensor-based methods for uncertain environments [3,8,17,20,35,51,60], when and what kind of virtual tools to use in our method completely depends on the user's decision. We implement virtual tools, including a point, line, and plane, and provide methods for smooth force rendering during the transition between the use of two consecutive tools in guidance. In addition, a virtual spring (also a kind of motion constraint) is connected to the force-reflection joystick and the robot manipulator, so that the linkage between the user and the robot can be established for better manipulation. For contact tasks, a virtual bumper attached on the robot manipulator for protection when interacting with stiff environments. For demonstration, the system is employed for the tasks of contour following and screw fastening, both of which involve delicate and challenging maneuvers. According to these two different design concepts, we name them as "environment-oriented motion constraint" and "user-oriented motion constraint," respectively.

Among various kinds of haptic devices [24,56], we choose the force-reflection joystick to develop our manipulation systems, due to the merits of simplicity and generality. Regarding the desirable features of force-reflection interfaces, Srinivasan and Basdogan [57] summarized three requirements: i) low back-drive inertia and friction, ii) ergonomics and comfort, and iii) matchable range, resolution, and bandwidth. And, according to [33,57], typical forces for humans used in exploration and manipulation are in the range of 5 to 15 N, and the spatial resolution of two points about 1 mm on the fingertip. For most of haptic systems, the force updating frequency of 1 kHz or more is demanded to provide a high quality of haptic rendering to the user [24,52,57].

1.2 Dissertation Contributions

Major contributions of this dissertation are briefly summarized below:

- We propose two design methods for environment-oriented and user-oriented virtual motion constraints. The former is applicable for well-known environments and the latter can be used to help the user when facing unstructured environments. We apply them to simulation systems for emulating the manipulative feeling and robot manipulation for assisting the user in governing, respectively, to verify their effectiveness.

- We develop a multi-functional virtual manipulation system that can be applied to a wider range of simulation systems, based on a 2-DOF force-reflection joystick and the proposed virtual motion constrains. The proposed virtual manipulation system has been experimentally verified via emulating an omni-directional wrench and a manual gearshift system, that can capture the main features of various manipulative devices.

- We present a systematic approach to design and implement the virtual motion constrains that correspond to the well-known simulated tasks, and also propose a pixel-based method for constructing these 2-D virtual motion constraints easily by using the graphic editing software, which can also achieve smooth force rendering during the transition between the walls.

- We develop an effective and natural manipulation system for the user to control a multi-DOF position-controlled robot manipulator via a 6-DOF force-reflection joystick in unstructured environments for contact tasks. The effectiveness of the proposed manipulation system has been experimentally verified by performing two kinds of tasks, contour following and screw fastening, which involve delicate and challenging maneuvers.

- We develop a set of virtual tools, which can provide position and orientation assistances to the user according to her/his call in real-time. According to our survey, the orientation assistance is less discussed in previous works. We also propose an algorithm to implement three types of position tools: a point, line, and plane.

1.3 Dissertation Organization

In this chapter, the motivation and contributions of this dissertation have been addressed. The rest of this dissertation is organized as: Chapter 2 describes the environment-orientated design method to tackle simulated tasks and its application in developing a multi-functional virtual manipulation system for various simulation tasks. In Chapter 3, we develop a set of virtual tools, based on user-oriented motion constraints, to assist the user for manipulating a multi-DOF robot manipulator in uncertain environments. Finally, the conclusions and future works are given in Chapter 4.

Chapter 2 Multi-Functional Virtual Manipulation System

In this chapter, we develop a multi-functional virtual manipulation system that can be applied to a wider range of simulation systems, based on using a 2-DOF force-reflection joystick, and thus propose a systematic approach to design and implement the virtual motion constraints that corresponds to various task requirements. To evaluate its effectiveness, we apply the proposed virtual manipulation system to emulate an omni-directional wrench and a manual gearshift system.

2.1 Introduction

Simulation systems have been applied to emulate practical systems, assist in mechanical design, and serve as the interface for teleoperation, among others [18,19,28,43]. With the development of virtual reality (VR) and dynamic modeling techniques, today's simulated environments are more realistic, and thus give the user a better feeling of immersion. And, via various communication channels, especially the bilateral manipulative devices, the user may interact with the simulation system in a more natural and efficient manner [12,13,30,43,44]. Due to the great variety of tasks that one may need to simulate, there is a strong demand to have a single manipulative device applicable for various tasks, which poses severe challenges.

In this chapter, we intend to design a virtual manipulation system that can be applied to a wider range of simulation systems, but not for exact replication of the real manipulative devices. Because our focus is on the compliance task, which requires both position and force management, we concentrate on the haptic device. Among various kinds of haptic devices, the force-reflection joystick, which has merit in its simplicity and generality, is chosen for our proposed system [9,45]. Our design adopts the concept of virtual mechanisms and virtual fixtures previously proposed [28,48,49,53]. The basic idea is to generate virtual motion constraints via the software, so that the joystick is constrained to move within a limited workspace that corresponds to task requirements. In other words, the virtual constraints are designed to make the joystick behave similar to the manipulative device it emulates, e.g., a steering wheel, wrench, or gearshift lever. Consequently, the users may feel as though they are operating a manipulative device that is specifically designed for the given task, thus achieving fast and effective manipulation.

We propose a systematic approach to design and implement the virtual motion constraints to tackle various kinds of compliance tasks. First, sets of virtual walls are developed to serve as the building blocks. Motion constraints for complex tasks can then be constructed via proper assembly of these basic building blocks. As different motion constraints may be required in different stages of compliance task execution, the virtual walls may be made of different shapes, placed in different orientations, and with different levels of hardness and stickiness. Furthermore, their shapes, orientations, and physical properties may vary along with the execution of the task. For instance, in a screw-driving task, the screw is tightened to the fixture gradually, with an increasing effect on friction during the process. We also propose a pixel-based method to achieve smooth force rendering between the connected walls. To evaluate whether the proposed virtual manipulation system can capture the main features of various manipulative devices, at the expense of exact replication, we apply it to emulate an omni-directional wrench and a manual gearshift system based on using a 2-DOF force-reflection joystick developed in our laboratory [37]. We then analyze the responses of the users during their manipulation. The rest of this chapter is organized as: Section 2.2 describes the proposed virtual manipulation system and how to construct the virtual motion constraints. Experiments for the constructions of both the virtual omni-directional wrench and virtual manual gearshift system are presented in Section 2.3. Section 2.4 provides the discussions on the evaluation of users' responses during the experiments. Finally, summaries are given in Section 2.5.

2.2 Proposed Virtual Manipulation System

Fig. 2.1 shows the conceptual diagram of the proposed multi-functional virtual manipulation system. The proposed virtual manipulative device consists of mainly a force-reflection joystick and the motion constraints. With the joystick guided to move within the walls of the motion constraints, the user may gain better manipulability when executing the compliance task. During the manipulation, the user imposes a force on the joystick to move around the motion constraints. In turn, this virtual manipulative device generates the motion command and sends it into the simulated environment. Meanwhile, the force and position feedbacks incurred during the interaction between the virtual manipulative device and the simulated environment are sent back to the user via the joystick and the visual display. With both the reflected haptic feeling and visual feedback, the user can then determine his/her next move.

To let the proposed virtual manipulation system applicable for a wide variety of compliance tasks, the proposed approach must be able to create virtual motion constraints systematically. We list some major considerations in motion constraint design for com-

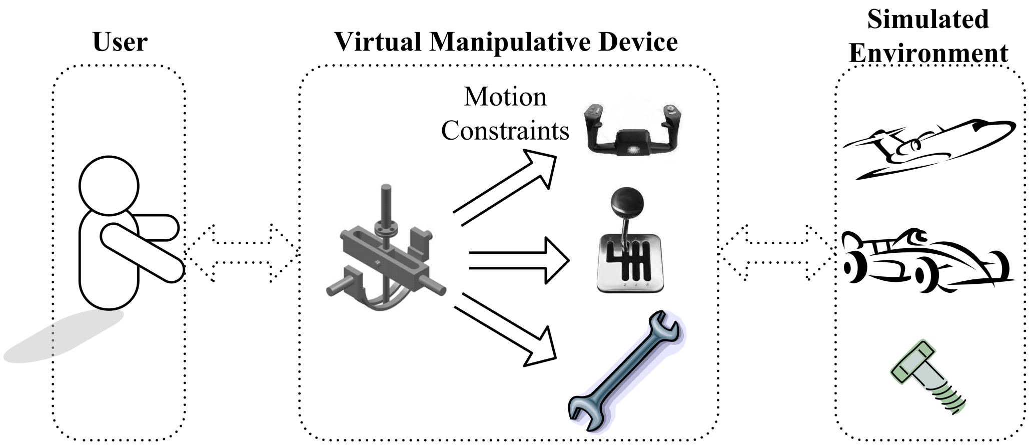


Fig. 2.1: Conceptual diagram of the proposed multi-functional virtual manipulation system.

pliance tasks below:

- For task formulation in the simulated environment, different shapes of motion constraints may be demanded and placed in different orientations. For instance, various virtual scenes may be present during the execution of a peg-in-hole task.
- Objects in various tasks may be made of different materials, and thus exhibit different physical properties.
- Along with the progress of the task execution, the shapes, orientations, and physical properties of the motion constraints may vary, e.g., the friction effect during the tightening of a screw into a fixture.

In responding to the requirements above, the virtual walls, as the building blocks for constructing the motion constraints, should provide varieties in their shape, orientation, and physical property. And, these attributes may be time-invariant or time-varying.

The design of these various kinds of virtual walls will be described in Section 2.2.1. In Section 2.2.2, we will discuss the connection of walls of various shapes and how smooth manipulation can be achieved when the joystick moves between walls. The procedure on how to construct the virtual motion constraints that correspond to task requirements will also be described.

2.2.1 Virtual Wall Design

During the design, we should first make the virtual wall behave like a real one, which implicates a demand of a very high stiffness for the virtual wall itself and an abrupt stiffness change during contact with the wall [14,39]. In addition, digital implementation of the virtual wall may be affected by a fast-moving joystick and low system sampling rate, which would result in deferred force generation [14]. This phenomenon may invoke system instability and cause the human operator to feel unnatural. To provide an instant reflective force, it may demand a sampling rate of up to more than 1 kHz [7]. Another issue in the design is about the pushing force a human operator may impose on the joystick. According to [59], a human hand may generate a pushing force up to above 60 N. If the operator keeps pushing against the joystick with the maximum force, he/she may bend the joystick and make the virtual wall collapse. These requirements have to be well tackled in implementing a realistic virtual wall.

In response to the requirements, we first constructed a 2-DOF force-reflection joystick that achieves the required sampling rate. Meanwhile, we also suggest that the joystick should be constrained to move slowly within the walls to alleviate the risk of a severe collision. The strategy proposed is to pair the virtual walls for forming the virtual aisles,

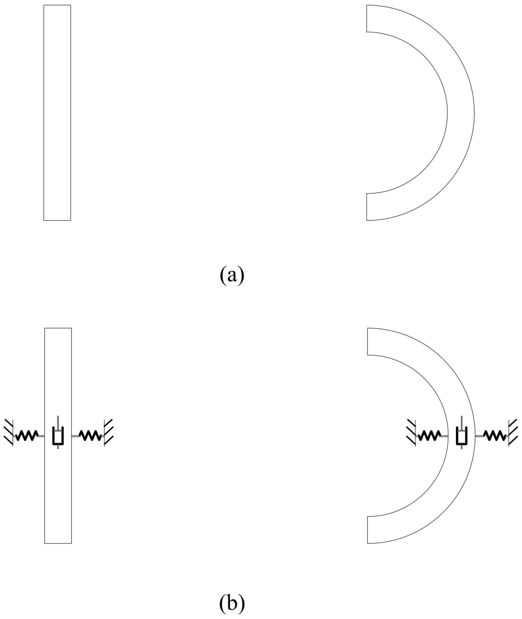


Fig. 2.2: Virtual aisle design: (a) rectangular and circular aisles and (b) physical property emulation.

so that the limited space should confine the movement of the joystick. We also assume that the human operator is aware that the virtual motion constraint serves as the guidance and reminder during the manipulation, and is expected not to push hard on the wall intentionally.

Following the concept above, we start with the rectangular and circular aisles as examples, as shown in Fig. 2.2(a). These aisles are designed to generate the reactive force responding to the push of the joystick during its manipulation within the aisle. As the joystick may touch the wall of the aisle in either a perpendicular or slanting direction, the reactive-force generation should depend on both its penetrating direction and depth inside the wall. Two common approaches for this are point-based and ray-based force rendering [57]. The former computes the force based on the stiffness provided by the joystick system and the distance between the collision point inside the wall and the nearest surface point (NSP); the latter uses the distance between the collision point and the surface point along the penetrating ray. We adopt the point-based approach because it yields more smooth force rendering. Force rendering for these two kinds of aisles can be easily extended to other shapes. Furthermore, to deal with various compliance tasks, physical properties in addition to the stiffness, represented by the spring in Fig. 2.2(b), will also be emulated. For instance, the friction effect for tasks like the tightening of a screw can be realized by installing the damping within the aisle, represented by the damper in Fig. 2.2(b). Damping can also be incorporated into the wall along with the stiffness to alleviate the possible oscillation due to hard hits. The reactive force generated during contact with the wall can thus be formulated as

$$f = K(x_{nsp} - x) - B\dot{x}, \tag{2.1}$$

where K stands for the stiffness, B the damping, and x_{nsp} the NSP corresponding to the current joystick position x.

2.2.2 Motion Constraint Construction

In assembling the walls for motion constraint construction, we first deal with the connection between them and the subsequent force rendering. Fig. 2.3 shows several types of connections between walls of the same or different shapes (again with rectangular and circular types of walls as examples). To smooth out the sharp corners between walls, the spline functions or others may be introduced for wall connection. When the joystick cruises through the vicinity of the connected area, smooth reactive force rendering is much more complicated than that with one single wall. Because a more complex shape

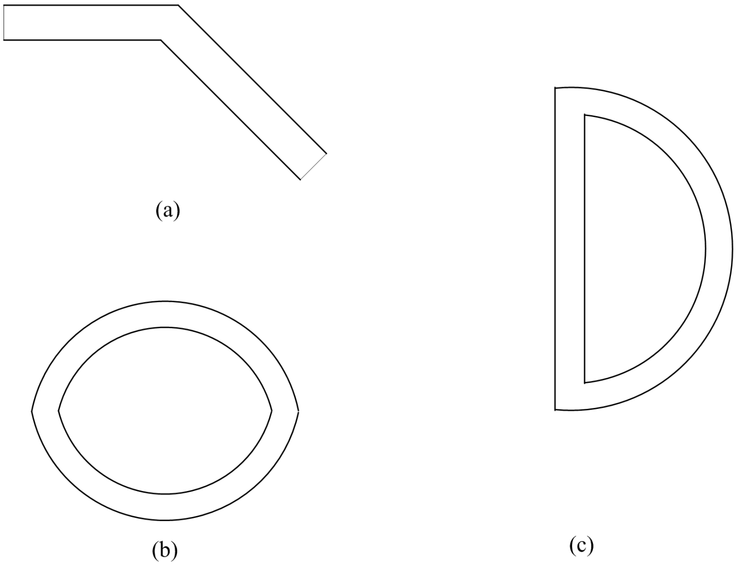


Fig. 2.3: Connection between walls: (a) line-to-line, (b) curve-to-curve, and (c) line-to-curve.

is formed, it may not be that straightforward to determine the direction of the reactive force [56]. One resolution for this is to keep tracking previous surface points, like the god-object and proxy algorithms previously proposed [54,64]. Motivated by this idea and to avoid describing connections of walls of various (even irregular) shapes with geometric equations case by case, we propose a pixel-based method for force rendering. In the proposed method, the interaction between the virtual joystick and motion constraints, the recording of the previous surface points, and the force rendering are all executed in the pixel space. Consequently, motion constraints can be easily constructed using the graphics editing software. Detection for contact between objects also becomes very straightforward. In the developed system, the resolution and updating rate for visual rendering are 0.4 mm between two neighboring pixels (500x500 pixels/200x200 mm^2) and 30 Hz, respectively, and those for haptic rendering are 0.1 mm (2000x2000 pixels/200x200

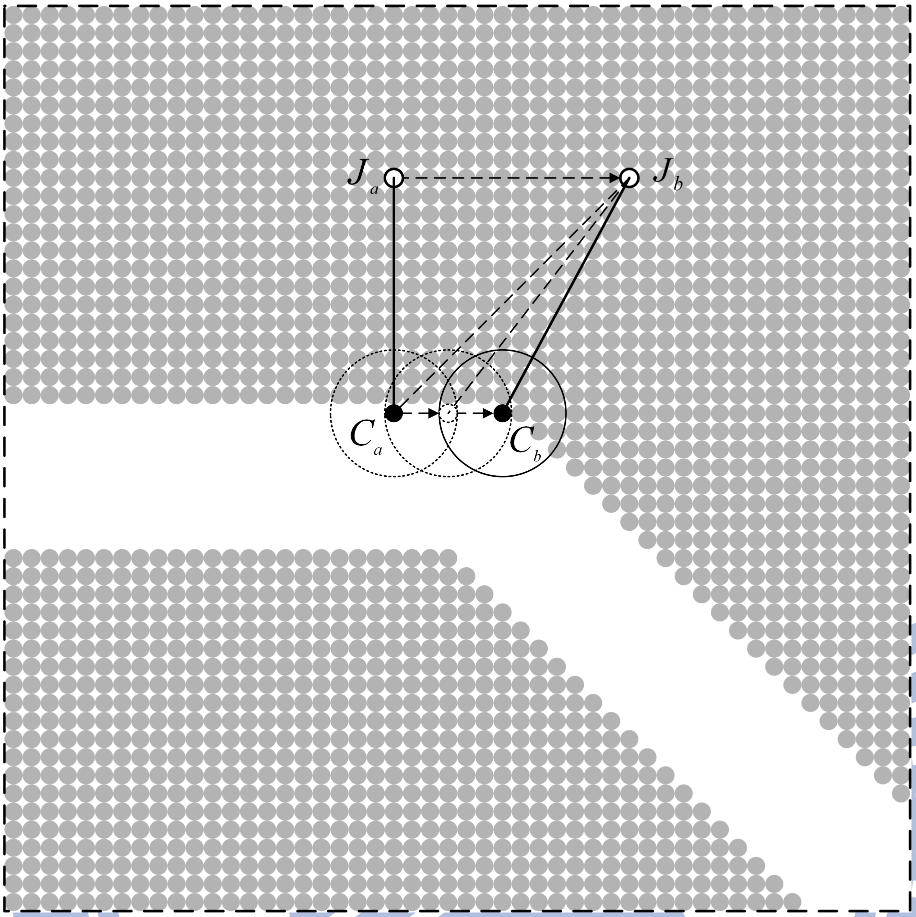


Fig. 2.4: Applying the proposed pixel-based method for force rendering for wall connection in Fig. 2.3(a).

mm^2) for each count and above 1 kHz, both satisfying the demand from the operator.

Fig. 2.4 shows an example of how the proposed method is applied for the case in Fig. 2.3(a), a connection of two rectangular aisles. In Fig. 2.4, the joystick is moving across the connected area from location J_a to J_b. With the point-based approach, the corresponding NSP for J_a is first identified as C_a. Next, the proposed method will search through a series of circular regions of a small radius to locate the possible NSP for J_b. The process will continue until the desired NSP C_b is found. Details of the searching process for locating C_b are given in Algorithm 1. Due to the small distance between each

Algorithm 1 Locating NSP C_b by C_a and J_b

```
1:  for $C_a$, find all $P_i \in \|P_i - C_a\| < \epsilon$, $i = 1, 2, ..., N$.
2:  for $i = 1$ to $N$ do
3:      if $P_i$ is a wall pixel then
4:          go to 2:
5:      end if
6:      if $\|P_i - J_b\| < \|C_a - J_b\|$ then
7:          $C_a \Leftarrow P_i$
8:          go to 1:
9:      end if
10: end for
11: $C_b \Leftarrow C_a$
```

step resulting from the high sampling rate, the computation load is not demanding. With these NSPs, smooth force rendering with a certain degree of continuity can be achieved. The user may thus experience a smooth maneuvering through the connected portion. In addition, this pixel-space implementation also has merit in direct quantization in the pixel space and extensibility to the 3D cases.

Procedure for motion constraint construction: Construct proper motion constraints corresponding to the requirements of a given compliance task.

Step 1: Perform analysis and specify the requirements for a given compliance task.

Step 2: According to the task requirements, select virtual walls of suitable shapes and physical properties.

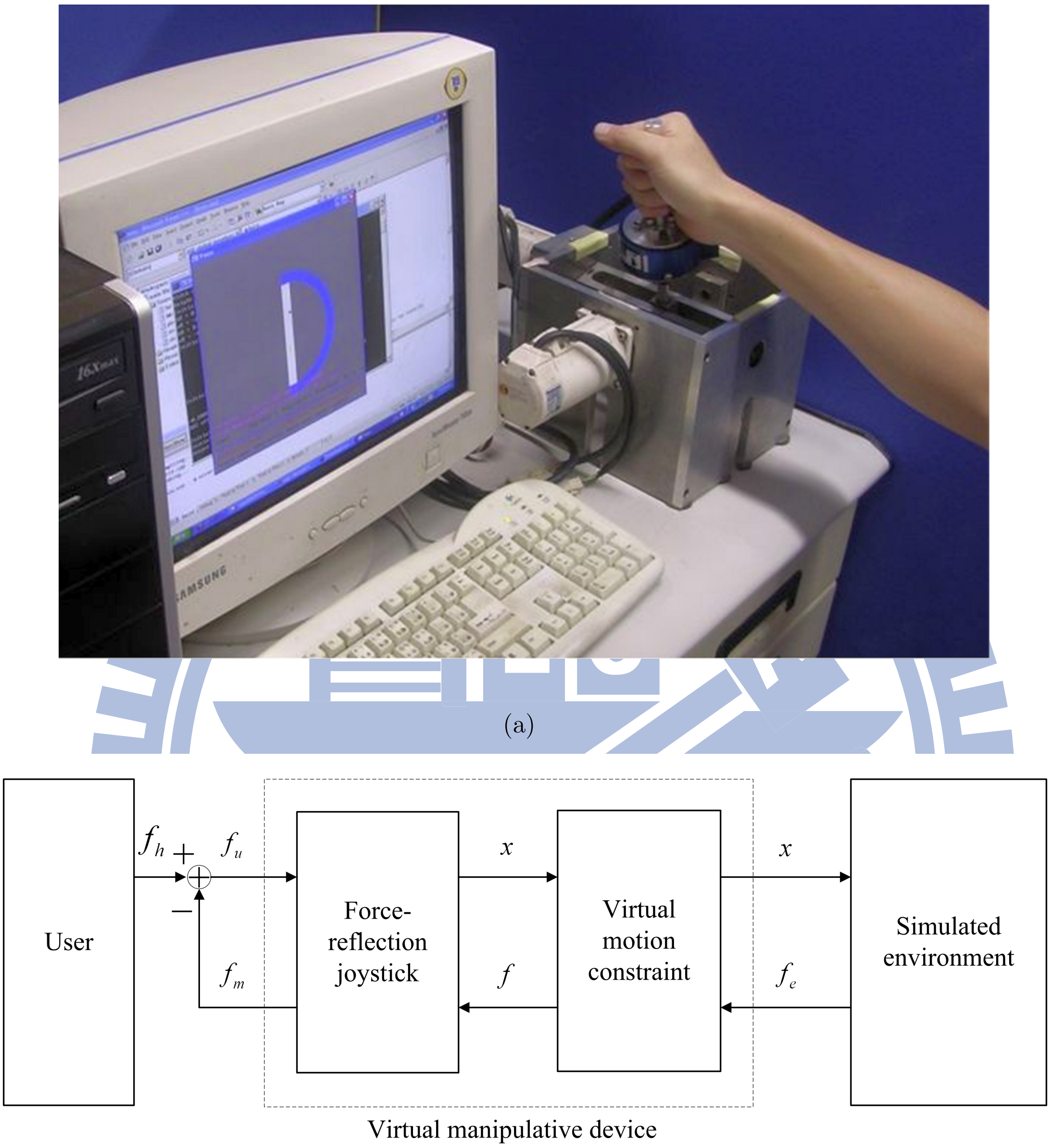
Step 3: Assemble the walls into aisles and proceed with the necessary connections between aisles. Perform force rendering for smooth manipulation using the proposed pixel-based method, when the joystick cruises through the connection.

Step 4: Along with the progress of the task execution, update the virtual constraints via proper adjustment of the virtual walls and their physical properties.

2.3 Experiments

During the experiments, we apply the developed multi-functional virtual manipulation system to emulate two kinds of manipulative devices, the omni-directional wrench and manual gearshift system. The purpose is to show that this manipulation system is able to capture the main features of several manipulative devices via a systematic approach, while at the expense of exact replication. The emulations of these two manipulative devices pose different requirements in implementation for involving both the circular and rectangular aisles and the physical properties in stiffness and damping.

Six male and two female graduate students from our institute, between 23 and 28 years old, were invited by our laboratory to perform the experiments. All of them had experience in manipulating the wrench and manual gearshift system. We developed a 2-DOF force-reflection joystick as the manipulative device, as shown in Fig. 2.5(a). This joystick is basically modified from our previous one [37] by using a peripheral with a higher sampling rate, whose specifications are listed in Table 2.1. The AC servo motor is used as the actuator to move the joystick directly. The motor (type MSMA041A1E, Panasonic, Japan) weighs 1.6 kg with a maximum speed of 4500 rpm and maximum output torque of 3.8 Nm. An encoder is installed within the motor to achieve closed-loop servo control with


User
f_h
+
−
f_u
f_m
Force-reflection joystick
x
f
Virtual motion constraint
x
f_e
Simulated environment
Virtual manipulative device

(a)

(b)

Fig. 2.5: (a) System view of the 2-DOF force-reflection joystick system developed in our laboratory. (b) System block diagram with the operator interacting with the simulated environment using the force-reflection joystick.

a resolution of 10000 pulses per revolution. The encoder is used to measure the position of the joystick. The resolution of the encoder and the quality of the actuator's servo control together determine the resolution of the joystick. For force control, the motor driver

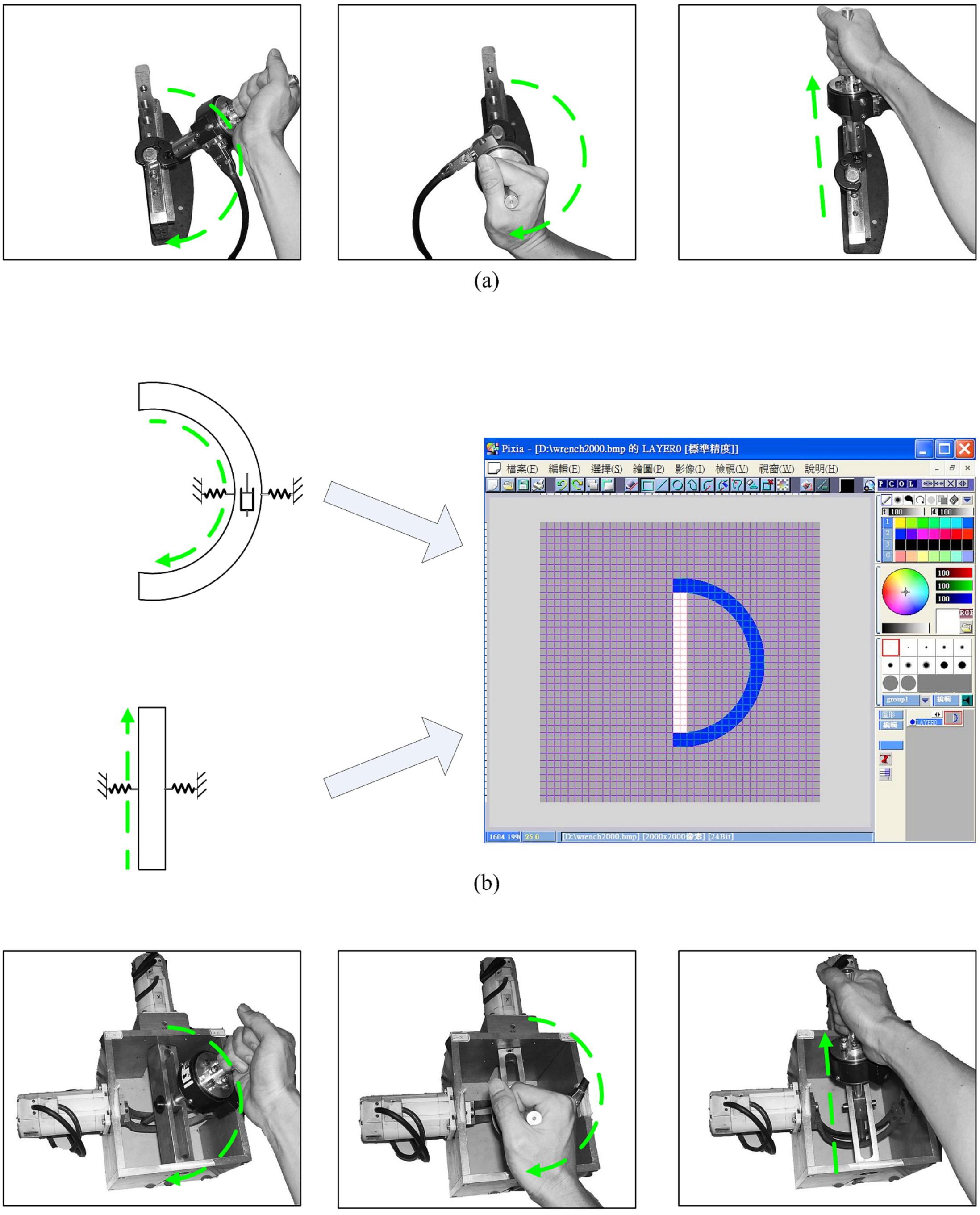
Table 2.1: Specifications of the developed 2-DOF force-reflection joystick.

Item	**Specification**
X axis	± (70 ∼ 80) deg
Y axis	± (80 ∼ 90) deg
Stick length	22 cm
Maximum output force	32.5 N
Precision	0.1 mm/count (28 counts/deg.)
Interface	PCI
Sampling rate	above 1 kHz
Bandwidth	100 Hz

is set at torque control mode. And, system identification has been performed via the frequency-response analysis. Fig. 2.5(b) shows the system block diagram when the user operates the joystick, as a virtual manipulative device, to interact with the environment. In Fig. 2.5(b), the user sends in the commanded force f_h and receives the reflection force f_m from the joystick. f_h and f_m, combined to be f_u, are sent to the joystick, which in turn generates the command x. Via the virtual motion constraint, x is forwarded to the simulated environment, which responds with an interactive force f_e. This f_e becomes f and then f_m through the virtual manipulative device.

2.3.1 Virtual Omni-Directional Wrench

We first apply the proposed manipulation system to construct a virtual omni-directional wrench. Fig. 2.6 shows how to construct the motion constraints for the virtual omni-



Pixia - [D:\wrench2000.bmp 的 LAYER0 [標準精度]]
檔案(F) 編輯(E) 選擇(S) 繪圖(P) 影像(I) 檢視(V) 視窗(W) 說明(H)
1604 1991 25.0
[D:\wrench2000.bmp] [2000x2000像素] [24Bit]

(a)

(b)

(c)

Fig. 2.6: Motion constraint construction for the virtual omni-directional wrench: (a) manipulation of the real omni-directional wrench, (b) virtual motion constraint generation, and (c) manipulation of the virtual omni-directional wrench.

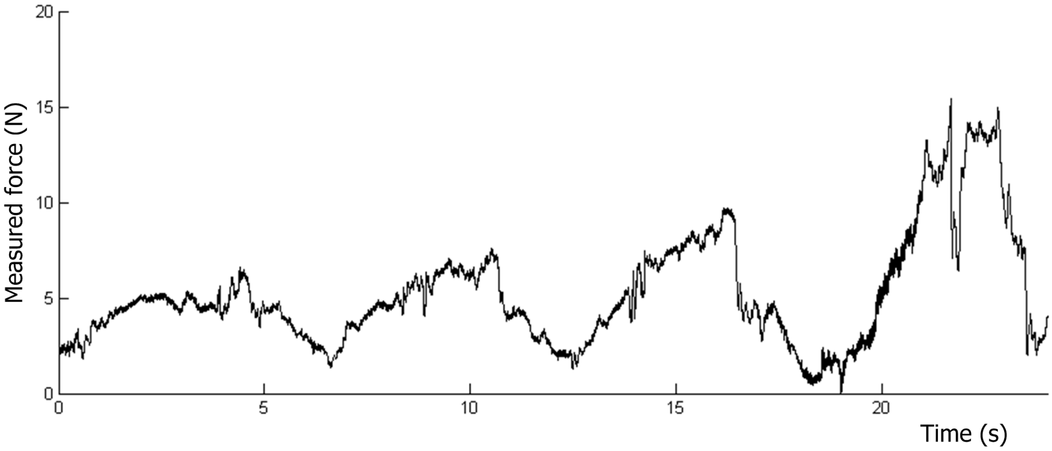


Fig. 2.7: Measured force response for screw tightening using a real wrench.

directional wrench based on manipulating a real wrench for a screw-tightening task. In Fig. 2.6(a), the user turns the wrench every one-half circle clockwise to tighten the screw, and then moves it back to the original location via a linear path. In response to these two motions, one semi-circular and one rectangular virtual aisle are generated, as shown in Fig. 2.6(b). Both of them are equipped with the springs on the two sides of the aisle to provide the resistive force, but only the semi-circular one is equipped with the damper inside of it to emulate the tightening effect. Note that the vertical movement of the screw during wrench turning is neglected due to its smallness. To make the virtual wrench approximate the real one more closely, we actually manipulated the real wrench to tighten a screw. Fig. 2.7 shows the force response measured by the JR3 force sensor (UFS-3012A-25, NITTA, Japan) mounted on the real wrench shown in Fig. 2.6(a). The measured forces, ranging between 0-16 N, increased gradually along with the turning of the wrench, as indicated by the four peaks in Fig. 2.7. Based on the observation, the damping coefficient B_w, used for exhibiting the increasing friction effect, is thus formulated to be proportional to the screw turning angle α:

$$B_w = \begin{cases} c\alpha & if\ \alpha \geq 0 \\ 0 & if\ \alpha < 0, \end{cases} \tag{2.2}$$

where c is a ratio. This process can be viewed as that for friction model identification. With these motion constraints, the joystick, confined to move within the aisles, behaves like a wrench, as shown in Fig. 2.6(c).

The virtual wrench was applied to execute the screw-tightening task in Fig. 2.6. During the first experiment, the user turned the wrench clockwise four times in tightening the screw. To maintain a constant speed in turning, a guiding ball (designed to be small for less distraction) was presented to visually lead the user to pass through the semi-circular and rectangular aisles in 6 seconds. This slow, constant movement is intended for the purpose of better observation and performance comparison. And, the ratio c for damping was set to be 5 N·s/m·rad. The JR3 force sensor was mounted on the handle of the joystick to measure the force reflected on the operator's hand. The resultant torque can be derived by taking the stick as the lever, with an effective length of 15 cm, as listed in Table 2.1. Fig. 2.8(a) shows the screw-tightening process, in which the wrench went through four turns of tightening, Fig. 2.8(b) the trajectory of the wrench (in angle), and Fig. 2.8(c) the measured force response, which increased gradually along with a steady turning wrench. In the second experiment, the user first tightened the screw and then loosened it by turning the wrench clockwise four times, followed by four counterclockwise turns. We also increased the ratio c up to 15 N·s/m·rad to check on the effect of the damping. Fig. 2.9(a) shows both the tightening and loosening processes, Fig. 2.9(b) the trajectory of the wrench, and Fig. 2.9(c) the measured force response, which increased gradually during the tightening process, and decreased gradually during that of loosening.

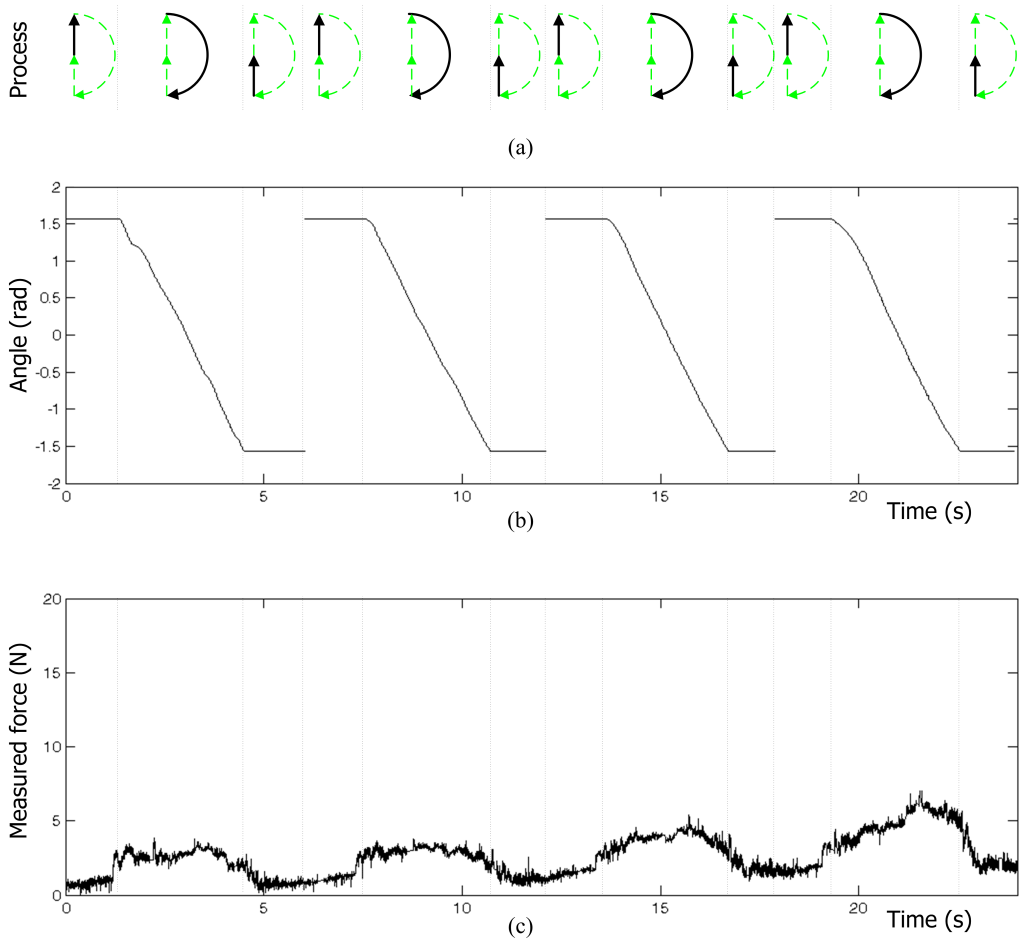


Fig. 2.8: Experimental results for applying the virtual omni-directional wrench for screw tightening ($c = 5$ N·s/m·rad): (a) the screw-tightening process, (b) the trajectory of the wrench (in angle), and (c) the measured force response.

Compared with that in Fig. 2.8(c), a more evident force variation was observed during the process.

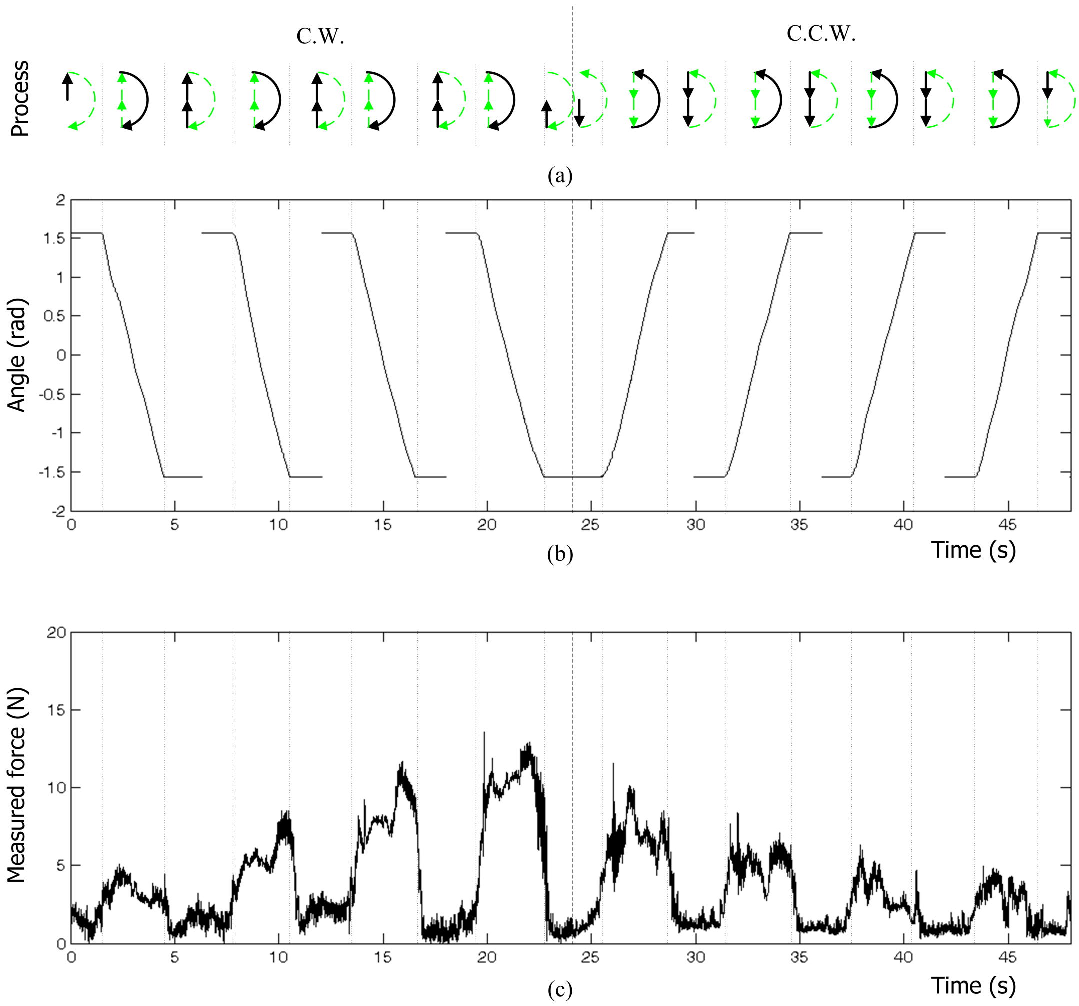
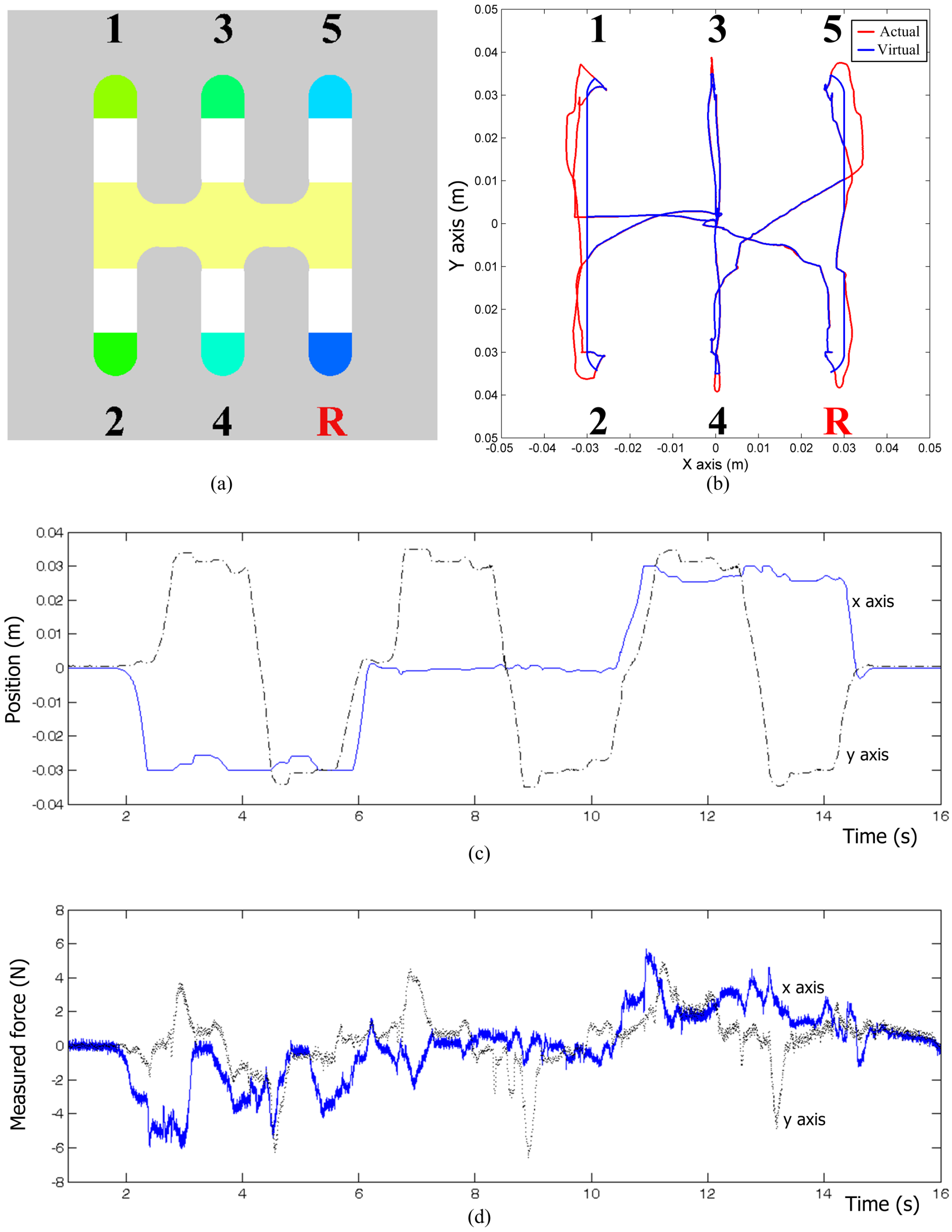


Fig. 2.9: Experimental results for applying the virtual omni-directional wrench for both screw tightening and loosening (c = 15 N·s/m·rad): (a) the screw-tightening (clockwise) and loosening (counterclockwise) processes, (b) the trajectory of the wrench (in angle), and (c) the measured force response.

2.3.2 Virtual Manual Gearshift System

As another demonstration, we applied the proposed system to construct a virtual manual gearshift system, and let the 2-DOF force-reflection joystick emulate a gearshift lever. Fig. 2.10(a) shows the developed virtual manual 5-speed gearshift system, which consists



(a)

(b)

(c)

(d)

Fig. 2.10: Experimental results for the emulation of a virtual manual 5-speed gearshift system: (a) the virtual manual 5-speed gearshift system, (b) the position trajectory of the virtual gearshift lever, (c) the position responses along the X and Y axes, and (c) the measured force response.

of one horizontal and three vertical aisles. The five speed gear positions and that for the reverse gear are located at the two ends (semi-circular walls) of the three vertical aisles, marked by 1 to 5 and R, respectively. Our design also includes a locking function when the joystick rests in the six gear positions, and an automatic return to the neutral position, located at the center of the horizontal aisle, when the joystick is away from the gear positions. Fig. 2.10(b) shows one demonstrated position trajectory of the virtual gearshift lever (blue line), in which the lever started from the neutral position, moved to the five speed gear positions and the reverse gear position, and finally returned to the neutral position. And in Fig. 2.10(b), the red line shows the actual position of the joystick while it hitting into the wall. Fig. 2.10(c) shows the corresponding position responses along the X and Y axes, respectively, and Fig. 2.10(d) the measured force response, which reflected how the joystick hit the wall during the movement (a larger force indicated a harder hit to the wall). During the manipulation, the lever was confined to move within the aisles via the guidance provided by the virtual constraints.

2.4 User Response Evaluation

To evaluate the responses of the eight users, they were asked to score the degree (from 1 (lowest) to 5 (highest)) with which he/she agrees with some statements related to the manipulation during the experiments.

For the omni-directional wrench emulation, the statements are

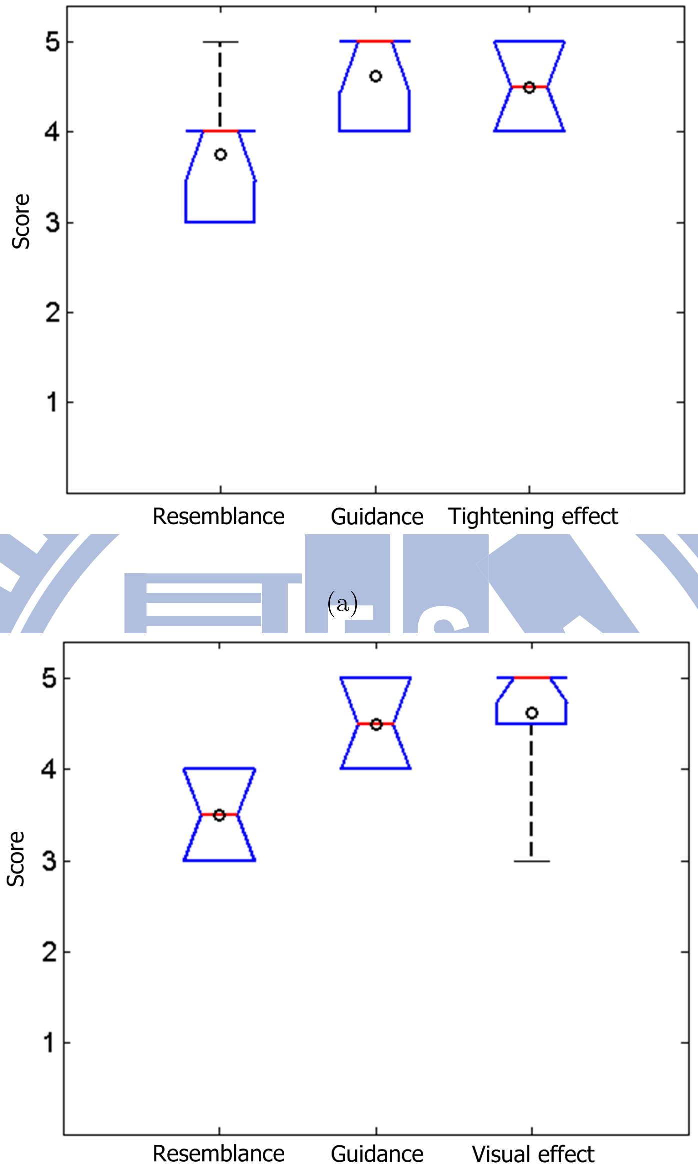
- I experience a realistic feeling as that of manipulating a real omni-directional wrench (Resemblance).

- I feel that the virtual constraints do provide the guidance (Guidance).
- I experience a realistic feeling of a screw-tightening process (Tightening effect).

For the manual gearshift system emulation, we asked the first two statements above, along with a third statement:

- I feel the visual information from the virtual environment is helpful during the manipulation (Visual effect).

Fig. 2.11(a) shows the scores for the omni-directional wrench, represented in box plots, and Fig. 2.11(b) those for the manual gearshift system. In Fig. 2.11(a), the box plot shows the maximum at 5, minimum 3, average 3.8, and median 4 for the resemblance statement, and higher scores for the guidance statement (average at 4.6 and median 5) and tightening effect statement (average and median both at 4.5). In Fig. 2.11(b), it shows the maximum at 4, minimum 3, and both average and median at 3.5 for the resemblance statement, and also higher scores for the guidance statement (average and median both at 4.5) and visual effect statement (average at 4.6 and median 5). From the scores, almost all the users felt that the virtual constraints did provide them the guidance during the manipulation for both emulations. Most users experienced a realistic feeling of a screw-tightening process, and considered the visual information was helpful for the gearshift system emulation. Meanwhile, several users considered the proposed manipulation system did exhibit certain degree of resemblance between the virtual and real manipulative devices in both emulations, although some differences were noticed. Judging from the evaluation results, we considered that the proposed manipulation system did capture the main features of these two manipulative devices, and yielded the realistic



(a)

(b)

Fig. 2.11: Evaluation of the users' responses: (a) the omni-directional wrench and (b) the manual gearshift system.

feeling to a certain extent.

The users also gave several comments that deserved attention. For both emulations, many of them mentioned that the spring effect appeared to be more impressive, when compared with that of the damping, which was not very realistic. Suggestions for improvement include the handle of the joystick should be modified to be more like that of the real gearshift lever, and the visual presentation of the interacting wrench and screw be provided. For the gearshift system emulation, the design of the attractive force for locking the lever into the gear position was appreciated.

2.5 Summary

In this chapter, we have proposed a systematic approach to design and implement the virtual motion constraints for a multi-functional virtual manipulation system based on a 2-DOF force-reflection joystick. Virtual walls of various shapes and physical properties are employed for motion constraint construction. And, a pixel-based method for force rendering is proposed to achieve smooth manipulation between walls, which well tackles walls of various shapes. The proposed system has been applied to emulate an omni-directional wrench and a manual gearshift system. And, the responses of the users during the manipulation have been analyzed. Although the proposed system is designed to be versatile, but not for exactly replication of the real mechanism, both emulations exhibit certain degree of resemblance and reality. For its extension to 3D applications, the virtual walls will need to be assembled in the 3D space. It thus induces higher complexity for motion constraint construction. Meanwhile, the proposed pixel-based method is ready for 3D force rendering as its implementation is in the pixel space and can be easily extended to 3D cases.

Chapter 3 Effective Manipulation for a Multi-DOF Robot Manipulator

In this chapter, we develop a set of virtual tools, based on the virtual motion constraints, to assist the user in natural and effective governing a multi-DOF robot manipulator for 3D applications with the requirements on both position and orientation via a 6-DOF force-reflection joystick. For demonstration, the proposed manipulation system is applied for the tasks of contour following and also screw fastening, both of which demand delicate maneuvers.

3.1 Introduction

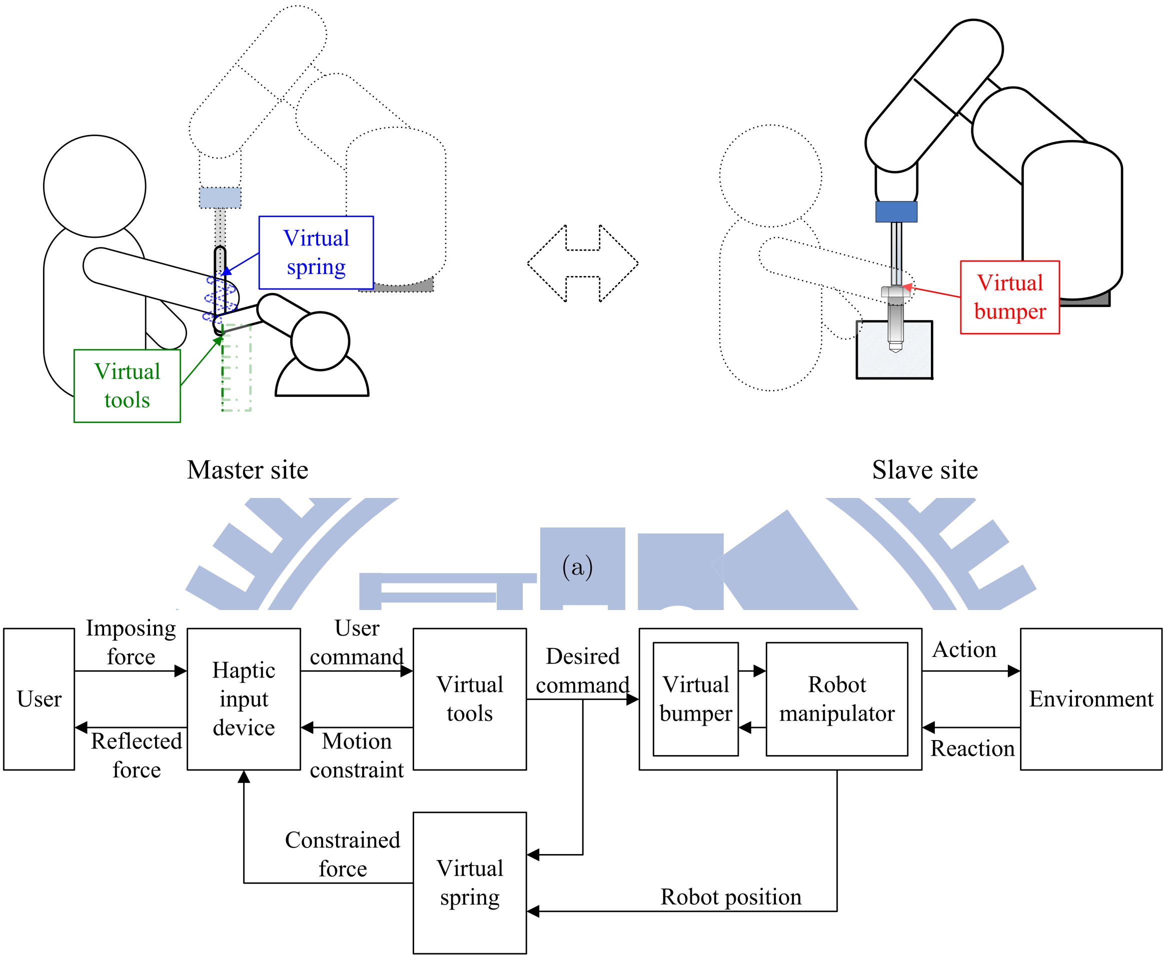
More robots are expected to enter human societies in the near future, as technologies in sensing, control, and artificial intelligence are in much progress. Instead of the organized environments, like factories, where most robots are currently deployed, they may need to face the uncertain and unstructured environments in homes, offices, laboratories, and others [29]. To tackle the high complexity present, we can imagine it would take too much effort to offer detailed programs to respond to all possible situations. One appealing alternate is to provide an effective manipulation system for the human to operate the robots, which motivates this study [30,47,55,58]. We choose a 6-DOF robot manipulator as the target robot to govern, as it is versatile and poses certain challenge for manipulation. For its popularity in practice, the robot manipulator is also taken to be under position control. Before the development of the proposed manipulation system, we have in-depth evaluated those manipulative devices commonly used, like teach box, mouse, keyboard, and joystick. As expected, they did not yield natural and efficient governing for a multi-DOF position-controlled robot manipulator. In addition, among previous researches on robot manipulation, several intuitive manipulative devices have been proposed, such as camera-based devices [46] and tangible user interfaces, like Wiimote [22], which recognize user's gestures for robot governing. They usually do not provide intuitive force feedback though. Some proposed using an identical robot as the user interface at a high cost [21].

The proposed manipulation system is designed to let the user feel she/he is well in control of the robot manipulator, although it is actually teleoperated. It consists of mainly a 6-DOF force-reflection joystick as the manipulative device and a set of virtual tools, along with a virtual spring and bumper, to assist the user for manipulation. A joystick with force reflection is employed for providing the haptic feedback and mutual interaction, which enhance the linkage between the user and robot manipulator [9,15,37]. The virtual tools are realized as the virtual motion constraints, based on the concept of virtual mechanisms and virtual fixtures previously proposed [6,28,49,53], to confine the movement of the manipulative device in the 3D working space, so that the robot manipulator can be properly guided for task execution. The virtual spring is designed to help the user recognize the spatial deviation between her/his desired position and the end-effector via a haptic clue. This function is demonstrated to be effective for governing the position-controlled robot manipulator. And, the virtual bumper is utilized to protect the robot manipulator from excessive contact force, which is crucial for tasks involving force management, e.g., screw driving.

The virtual tools are for the guidance of the user's movement to respond to task requirements. For instance, a virtual ruler or compasses can help the user to move the robot manipulator along a straight or circular line. In addition to position guidance, a focus of most previous works, the proposed system also provides orientation assistance to help the user, e.g., maintaining the vessel in an upright angle. Because some force-reflection joysticks may not be equipped with torque feedback, we thus develop a torque-free method for orientation assistance. In addition, the virtual tools are selected for assistance according to the current status of task execution dynamically, but not in a predefined environment or predicted way [5,17,31,63]. Software implementation of the proposed manipulation system is executed in a 3D graphical environment with a real-time manner, which also provides the visual feedback. To demonstrate its effectiveness, the manipulation system is used to govern the robot manipulator to conduct two kinds of tasks: contour following and screw fastening, both of which demand delicate maneuver. Users of various backgrounds are invited for the experiments, and their responses analyzed. The rest of this chapter is organized as: Section 3.2 describes the proposed manipulation system, including the realization of virtual tools, spring, and bumper. System implementation is presented in Section 3.3. Section 3.4 discusses the experiments and performance evaluation. Finally, summaries are given in Section 3.5.

3.2 Proposed Manipulation System

Fig. 3.1(a) illustrates the three major goals of the proposed manipulation system for a multi-DOF manipulator: (i) establishing a linkage between the user and robot manipulator via the virtual spring, so that the user may feel she/he is manipulating the robot on site directly, (ii) providing a set of virtual tools that guide the hand movement of the user according to task requirements, and (iii) attaching a virtual bumper on the robot



(a)

(b)

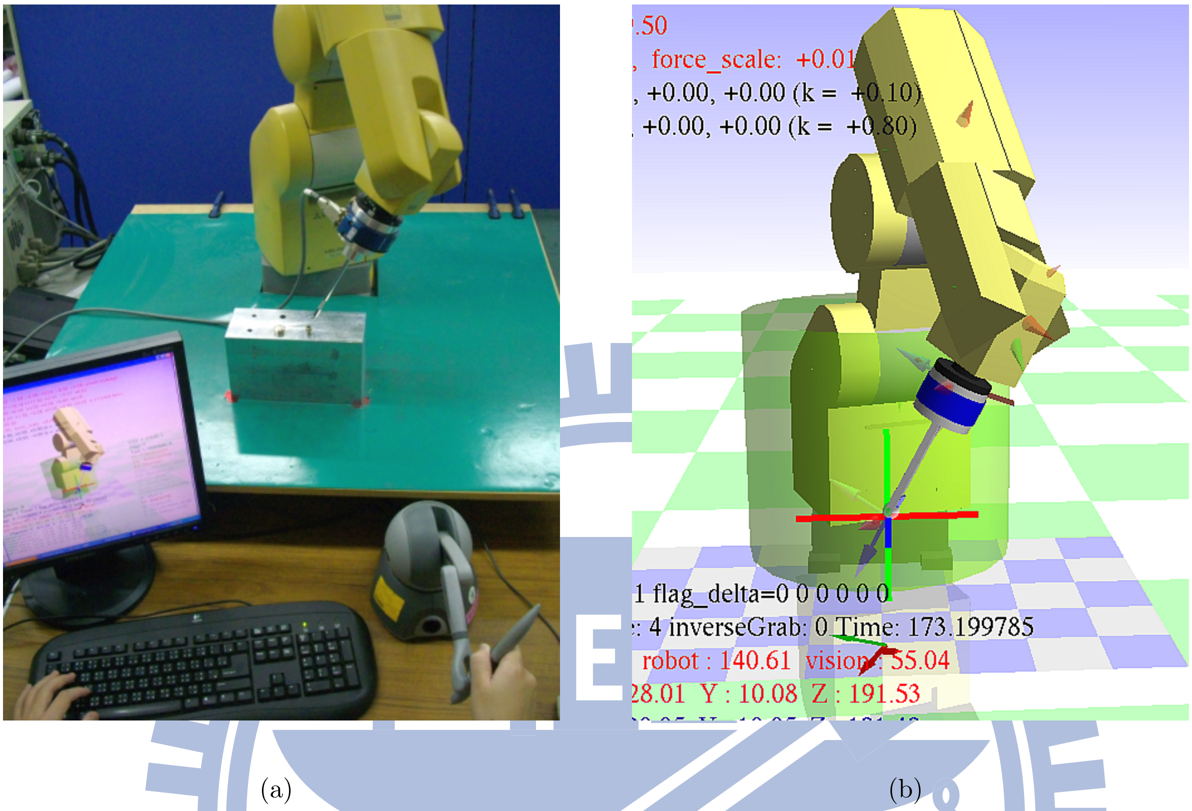
Fig. 3.1: Proposed manipulation system for a position-controlled robot manipulator: (a) conceptual diagram and (b) system block diagram.

manipulator for protection when interacting with the stiff environments. Fig. 3.1(b) shows its system block diagram, which includes a 6-DOF haptic device, a virtual spring, virtual tools, and a position-controlled robot manipulator with a virtual bumper. Via the haptic device, the user sends in both the position and orientation commands to govern the robot manipulator. These commands are modulated by the virtual tools, which utilize motion constraints for guidance. As the feedback, the positional deviation, via the virtual spring and then the haptic device, is transformed into the reflected force and sent back to the user to determine next move. Meanwhile, the virtual bumper would be activated to alleviate the contact force, when dealing with contact tasks. The success of the proposed manipulation system relies on proper realization of the virtual linkage, virtual tools, and virtual bumper. In Section 3.2.1, we describe how to establish the virtual linkage. In Section 3.2.2, we discuss how to implement the virtual tools for various types of tasks. And how to develop the virtual bumper is given in Section 3.2.3.

3.2.1 Virtual Linkage

With the proposed virtual linkage, we intend to create better interaction between the user and robot manipulator, so that the user can perform the manipulation naturally and effectively. Fig. 3.2 shows a scene in which the user applies the proposed manipulation system to manipulate the real robot manipulator via the virtual one. For the user to feel immersive in both visual and haptic senses, the proposed manipulation system employs the force-reflection joystick and virtual robot manipulator to provide the desired realism. For the visual sense, we utilize a 3D graphical system to build the virtual robot manipulator to be much alike the real one in both the appearance and motion. For the haptic counterpart, we intend to let the user feel like her/his hand holding on the end-effector of the robot manipulator directly. To achieve this haptic realism, we need to deal with (i) the alignment and workspace mapping between the coordinate systems of the joystick and robot manipulator and (ii) generation of constrained force that bonds the joystick with the robot manipulator.

We first deal with the alignment and workspace mapping between these two coordinate systems, shown in Fig. 3.3. We intend to find a transformation that maps position


.50
, force_scale: +0.01
, +0.00, +0.00 (k = +0.10)
, +0.00, +0.00 (k = +0.80)
1 flag_delta=0 0 0 0 0 0
: 4 inverseGrab: 0 Time: 173.199785
robot : 140.61 vision : 55.04
28.01 Y : 10.08 Z : 191.53

(a) (b)

Fig. 3.2: The user utilizes the manipulation system to manipulate (a) the real robot manipulator via (b) the virtual robot manipulator.

$\mathbf{p}_M$ and orientation $\mathbf{R}_M$ in the coordinate system of the joystick (centered at $\mathbf{c}_M$) into the corresponding position $\mathbf{p}_S$ and orientation $\mathbf{R}_S$ in that of the robot manipulator (centered at $\mathbf{c}_S$), with k as the scaling factor for matching up the size of the two workspaces. The transformations are formulated as

$$\mathbf{p}_S = k\mathbf{R}_M^S(\mathbf{p}_M - \mathbf{c}_M) + \mathbf{c}_S \tag{3.1}$$

and

$$\mathbf{R}_S = \mathbf{R}_M^S\mathbf{R}_M, \tag{3.2}$$

where $\mathbf{R}_M^S$ is the rotational matrix from the coordinate system of the joystick to that of the robot manipulator.

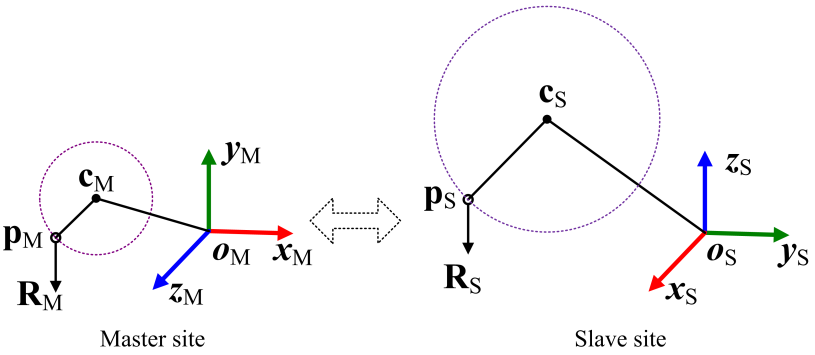
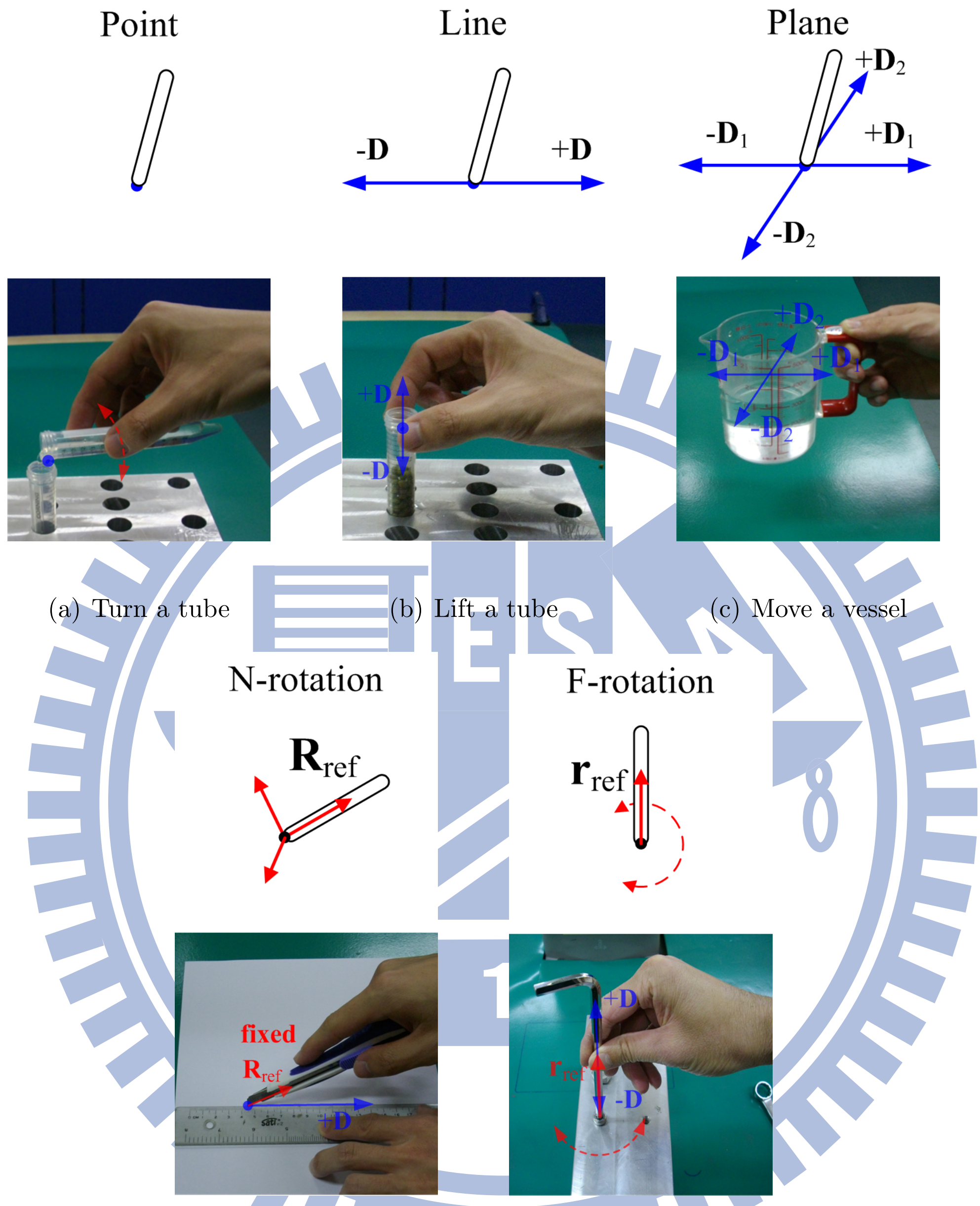


Fig. 3.3: Alignment and workspace mapping between the coordinate systems of the joystick and robot manipulator.

Although direct reflection of the environmental force to the master may yield the user a more realistic feeling, the induced master motion may make system unstable [32]. In addition, it is not preferred due to the high impedance of the position-controlled robot manipulator, which may induce large contact force when hitting a stiff environment. Therefore, the proposed linkage haptic cue is used to hint the positional deviation between the robot manipulator and the desired command, reminding the user that the robot manipulator cannot catch up with the command, in an intuitive and efficient way compared with that provided by the visual information. We then use a simple spring model to generate the constrained force via the force-reflection joystick, serving as a virtual spring that bonds the joystick with the robot manipulator. This constrained force $\mathbf{f}_s$ is formulated as

$$\mathbf{f}_s = K_s(\mathbf{p}_s - \mathbf{p}_u), \tag{3.3}$$

where K_s stands for the stiffness of the virtual spring, and $\mathbf{p}_s$ and $\mathbf{p}_u$ the position of the end-effector and user desired command, respectively, both with respect to the coordinate system of the master.



(a) Turn a tube (b) Lift a tube (c) Move a vessel

(d) Paper cutting (e) Turn a hex key

Fig. 3.4: Proposed virtual tools and their possible applications: (a) point, (b) line, (c) plane, (d) N-rotation, and (e) F-rotation.

3.2.2 Virtual Tools

To deal with general operations in daily lives, we have designed three types of position tools: point, line, and plane, and two types of orientation tools: N-rotation (non-rotation)

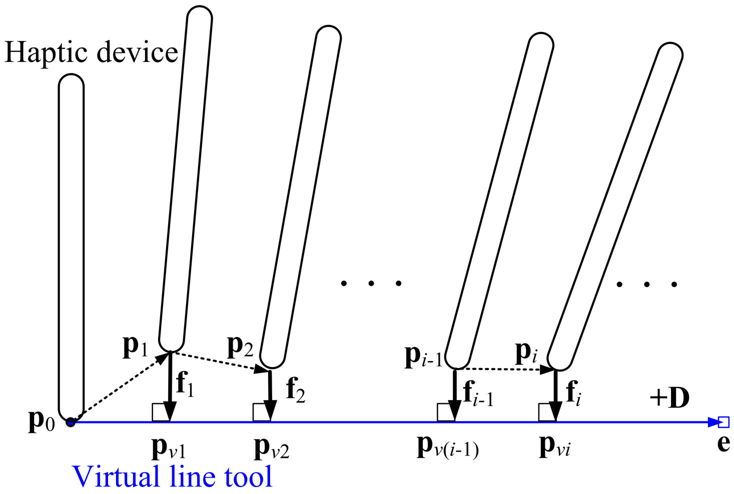


Fig. 3.5: The process of how the line tool executes motion assistance.

and F-rotation (fixed-axis-rotation), as shown in Fig. 3.4. In Fig. 3.4(a), the point tool confines the haptic device to rotate around a fixed point, and can be applied for tasks, like liquid pouring. The line and plane tools let the haptic device move along a straight line and within a plane, respectively, and can be applied for tasks, like lifting a tube and moving a vessel, as shown in Figs. 3.4(b) and 3.4(c). For orientation assistance, the N-rotation tool confines the haptic device to maintain a fixed orientation, i.e., it cannot be rotated. Fig. 3.4(d) shows an example of its application, in which it is used along with the point tool for paper cutting, and helps to let the knife maintain a fixed orientation with the paper during the cutting process. On the other hand, the F-rotation tool provides a fixed-axis rotation, so that it can be applied for tasks, like screw fastening, as shown in Fig. 3.4(e). As some tasks may have requirements on which direction to move, e.g., it demands the knife to move forward only during paper cutting, we also incorporate the directional constraints into the position tools.

We start with the design of the line tool. Fig. 3.5 illustrates how the the line tool executes motion assistance, in which the tool is orientated in a $+\mathbf{D}$ direction (later, we

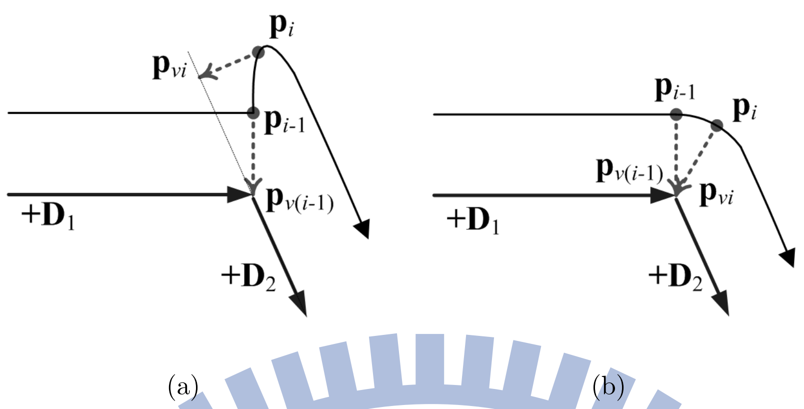


Fig. 3.6: Two force rendering methods for the transition of two consecutive tools: (a) the orthogonal-projection method (unsmooth force rendering) and (b) the proposed method (smooth force rendering).

denote such a forward-only line tool by $+\mathbf{D}$). In Fig. 3.5, when the haptic device deviates from the line specified by the line tool due to imprecise maneuver of the user, the line tool will generate a constrained force $\mathbf{f}_i$ to bring the tip of the haptic device $\mathbf{p}_i$ back to the line. This $\mathbf{f}_i$ is formulated as

$$\mathbf{f}_i = K_t(\mathbf{p}_{vi} - \mathbf{p}_i), \tag{3.4}$$

where K_t is the stiffness for the virtual tool to come up with a proper constrained force, and $\mathbf{p}_{vi}$ the nearest tool point (NTP) on the line tool that corresponds to $\mathbf{p}_i$. For some cases, it is not that straightforward to locate a proper NTP as that shown in Fig. 3.5. As an example, Fig. 3.6(a) shows the transition from tool $+\mathbf{D}_1$ at $\mathbf{p}_{i-1}$ to another $+\mathbf{D}_2$. If we only consider the orthogonal projection alone, $\mathbf{p}_{vi}$ for $\mathbf{p}_i$ will be located a the location as shown in Fig. 3.6(a) due to the presence of $+\mathbf{D}_2$, leading to unsmooth force rendering. To ensure smooth force rendering, we continue to use the strategy adopted in the pixel-based algorithm previously developed [26], but extend it for the 3-D case, which keeps tracking previous NTPs when locating the next one. Fig. 3.6(b) shows how the proposed

Algorithm 2 Line tool: Locating NTP $\mathbf{p}_{vi}$ by $\mathbf{p}_i$, $\mathbf{p}_{v(i-1)}$, and $+\mathbf{D}$ with $\mathbf{e}$

```
1: let $\mathbf{p}'_v = \mathbf{p}_{v(i-1)} + \delta(+\mathbf{D})$
2: if $(\mathbf{e} - \mathbf{p}'_v) \cdot (+\mathbf{D}) > 0$ then
3:     if $\|\mathbf{p}_i - \mathbf{p}'_v\| < \|\mathbf{p}_i - \mathbf{p}_{v(i-1)}\|$ then
4:         $\mathbf{p}_{v(i-1)} \Leftarrow \mathbf{p}'_v$
5:         go to 1:
6:     end if
7: end if
8: $\mathbf{p}_{vi} \Leftarrow \mathbf{p}_{v(i-1)}$
```

method works for the same case in Fig. 3.6(a). In Fig. 3.6(b), when the tool $+\mathbf{D}_2$ is called upon, the proposed method will search through a series of adjacent points, starting from $\mathbf{p}_{v(i-1)}$ in the $+\mathbf{D}_2$ forward direction to determine the NTP for $\mathbf{p}_i$. As a result, it locates the NTP $\mathbf{p}_{vi}$ still the previous $\mathbf{p}_{v(i-1)}$ for $\mathbf{p}_i$, leading to smooth force rendering. In Fig. 3.5, the line tool is orientated in the $+\mathbf{D}$ direction, for allowing only movement along this $+\mathbf{D}$ direction. And this line tool is also with an $\mathbf{e}$ point marked at its end, which is intended for some applications that prohibit the haptic device to move past a specified location. Based on the discussions above, we develop Algorithm 2, which locates the NTP, $\mathbf{p}_{vi}$, corresponding to $\mathbf{p}_i$ by previous $\mathbf{p}_{v(i-1)}$ and $+\mathbf{D}$ with $\mathbf{e}$. Algorithm 2 can be easily extended for the bidirectional line tool by taking $+\mathbf{D}$ for forward movement and $-\mathbf{D}$ for backward one. As for the plane tool, it will take two of those for the line tool, as its search of NTP will be in two degrees of freedom. And, for the point tool, the NTP is exactly the point selected for rotation. To satisfy the demand for haptic rendering, δ is set to be 0.1 mm and haptic updating frequency about 1 kHz.

We move on to design the two orientation tools: N-rotation and F-rotation. The

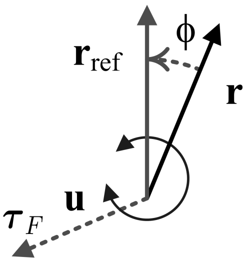


Fig. 3.7: Illustration for applying the F-rotation tool for assistance.

N-rotation tool is used to let the tool maintain a fixed orientation. When the orientation of the haptic device $\mathbf{R}$ deviated from the reference orientation $\mathbf{R}_{ref}$ due to imprecise maneuver, a constrained torque $\boldsymbol{\tau}_N$ is generated to rotate the haptic device back to $\mathbf{R}_{ref}$, formulated as

$$\boldsymbol{\tau}_N = K_N \theta \frac{\mathbf{v}}{\|\mathbf{v}\|}, \tag{3.5}$$

where θ is the deviated angle, $\mathbf{v}$ the rotation direction of torque $\boldsymbol{\tau}_N$, and K_N the stiffness for the N-rotation tool. Because the haptic device may not be equipped with torque feedback, the strategy adopted by this N-rotation tool is to let $\mathbf{R}_{ref}$ stay as the orientation command, no matter what the current $\mathbf{R}$ from the user is. In other words, the role of $\boldsymbol{\tau}_N$ is replaced by a fixed orientation command. Meanwhile, without the guidance via the torque feedback, like that from the force feedback used for the line tool, this strategy may induce abrupt rotation, once the N-rotation tool is released. To tackle it, the end-effector is not demanded to catch up with the haptic device after release immediately, but via a small velocity gradually.

The F-rotation tool is intended for providing a fixed-axis rotation. Fig. 3.7 illustrates how to apply the F-rotation tool for assistance, in which the rotating axis of the haptic device $\mathbf{r}$ deviated an angle ϕ from the reference $\mathbf{r}_{ref}$ due to imprecise maneuver. A constrained torque $\boldsymbol{\tau}_F$ is then generated to rotate $\mathbf{r}$ back to $\mathbf{r}_{ref}$, formulated as

$$\boldsymbol{\tau}_F = K_F \phi \mathbf{u} \tag{3.6}$$

with

$$\phi = \cos^{-1}(\mathbf{r} \cdot \mathbf{r}_{ref}) \tag{3.7}$$

and

$$\mathbf{u} = \frac{\mathbf{r} \times \mathbf{r}_{ref}}{\|\mathbf{r} \times \mathbf{r}_{ref}\|}, \tag{3.8}$$

where $\mathbf{u}$ is the rotation direction of torque $\boldsymbol{\tau}_F$ and K_F the stiffness for the F-rotation tool. Similar to the design for the N-rotation tool, the role of $\boldsymbol{\tau}_F$ is replaced by the orientation command $\mathbf{R}_v$, which allows the end-effector to rotate around the reference axis $\mathbf{r}_{ref}$. This $\mathbf{R}_v$ is formulated as

$$\mathbf{R}_v = \mathbf{R}_{\mathbf{u}}(\phi)\mathbf{R}, \tag{3.9}$$

where $\mathbf{R}_{\mathbf{u}}(\phi)$ is the rotation matrix to coincide $\mathbf{r}$ with $\mathbf{r}_{ref}$.

3.2.3 Virtual Bumper

The virtual bumper is introduced to protect the robot manipulator from excessive contact force when interacting with stiff environments, which adjusts the user's position command according to measured contact force [27,58]. The concept for the design is quite straightforward: stop the robot manipulator along the direction of contact when it encounters a large contact force, and decrease the forward-moving tendency when the contact force is not that large. We first select a threshold for the contact force, f_{th}, for which the robot manipulator can stand. When the magnitude of the measured contact force $\mathbf{f}_e$ is smaller than f_{th}, the user's position command $\mathbf{p}_u$ is adjusted to be $\mathbf{p}_b$:

$$\mathbf{p}_b = \mathbf{p}_u + \alpha \mathbf{f}_e, \tag{3.10}$$

where α is a factor to determine the ratio of the trade-off between the position command and contact force. Under this case, $\mathbf{p}_b$ may still move the robot manipulator forward, but in a slow, cautious manner. When $\|\mathbf{f}_e\|$ is larger than f_{th}, we let $\mathbf{p}_b$ be its previous value, which stops the robot manipulator from moving forward. Meanwhile, the effect of the virtual bumper is also transferred to the user via the virtual spring, as larger deviation is generated between the desired command and actual robot position. Based on the discussions, we develop Algorithm 3, which determines the adjusted command $\mathbf{p}_{bi}$ by previous $\mathbf{p}_{b(i-1)}$ and current $\mathbf{p}_u$ and also $\mathbf{f}_e$, listed below:

Algorithm 3 Virtual bumper: Determining the adjusted command $\mathbf{p}_{bi}$ by $\mathbf{p}_u$, $\mathbf{f}_e$, and $\mathbf{p}_{b(i-1)}$

```
let $\mathbf{p}'_b \equiv \mathbf{p}_u + \alpha \mathbf{f}_e$
if $\|\mathbf{f}_e\| \leq f_{th}$ then
    $\mathbf{p}_{bi} \Leftarrow \mathbf{p}'_b$
else
    let $\Delta\mathbf{p}'_b \equiv \mathbf{p}'_b - \mathbf{p}_{b(i-1)}$
    if $\Delta\mathbf{p}'_b \cdot \mathbf{f}_e \geq 0$ then
        $\mathbf{p}_{bi} \Leftarrow \mathbf{p}'_b$
    else
        $\mathbf{p}_{bi} \Leftarrow \mathbf{p}'_b + \|\Delta\mathbf{p}'_b\| \frac{\mathbf{f}_e}{\|\mathbf{f}_e\|}$
    end if
end if
```

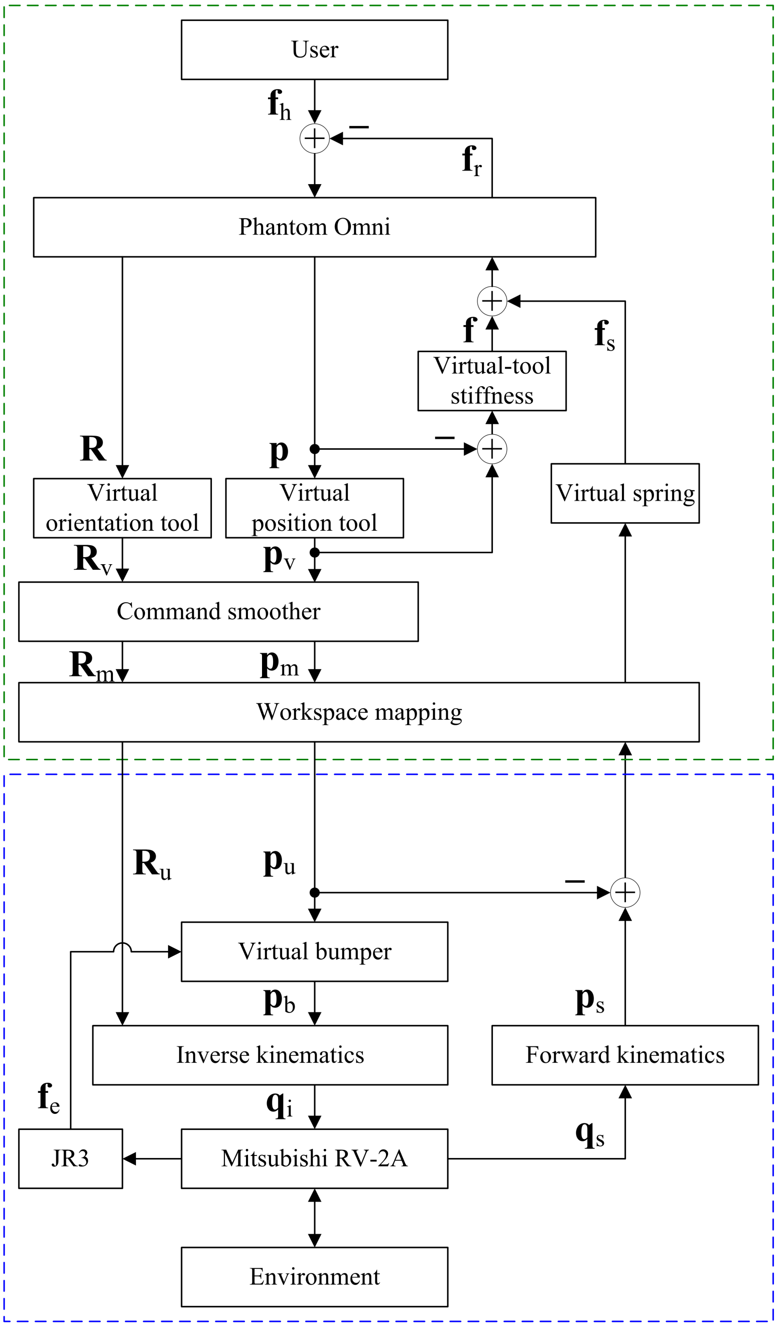


Fig. 3.8: System implementation block diagram of the proposed manipulation system.

3.3 System Implementation

Fig. 3.8 shows the system block diagram of the proposed manipulation system. The robot manipulator adopted is the Mitsubishi RV-2A 6-DOF robot manipulator (its joint specification listed in Table 3.1) with a 6-DOF force/torque sensor (JR3 IFS-90M31A25-I50-ANA) mounted on the end-effector. Its controller is type CR1-571 with a command cycle of 7.1 ms [40]. The 6-DOF SensAble Phantom Omni serves as the manipulative device, which provides 3-DOF force feedback (maximum force at 3.3 N), but not torque feedback [2]. A personal computer (Intel Core Duo E8400 CPU and 2 GB RAM) is used for computing haptic feedbacks from those virtual mechanisms and also related operations. In Fig. 3.8, the user sends in the command force $\mathbf{f}_h$ and receives the reflected force $\mathbf{f}_r$ from the Phantom Omni. This $\mathbf{f}_r$ represents the hatpic feedback resulting from the virtual position tool ($\mathbf{f}$) and virtual spring ($\mathbf{f}_s$). $\mathbf{f}_h$ and $\mathbf{f}_r$ together are sent to the Phantom Omni to generate corresponding position $\mathbf{p}$ and orientation $\mathbf{R}$. Via the virtual position and orientation tools, command smoother (discussed below), and workspace mapping, $\mathbf{R}$ and $\mathbf{p}$ become the user's commands in position $\mathbf{p}_u$ and orientation $\mathbf{R}_u$. For contact force alleviation, $\mathbf{p}_u$ is further modified by the virtual bumper to become the final position command $\mathbf{p}_b$. Via the module for inverse kinematics, joint solutions $\mathbf{q}_i$ corresponding to $\mathbf{p}_b$ and $\mathbf{R}_u$ is derived and sent to the Mitsubishi RV-2A for execution. As a feedback, the measured force $\mathbf{f}_e$ due to the interaction between the robot manipulator and environment is sent into the virtual bumper. And, joint $\mathbf{q}_s$, becoming position $\mathbf{p}_s$ via the forward kinematics module, is sent back to the manipulation system to initiate next operation.

The command smoother is designed to alleviate the undesired robot vibrations due to the unstable hand motion, which evidently affects the manipulation in certain cases [5]. In this command smoother, a sliding window of size N is used to find the average position $\mathbf{p}_m$ from previous N positions $\mathbf{p}_v$, formulated as

$$\mathbf{p}_{mi} = \frac{1}{N} \sum_{j=i-N+1}^{i} \mathbf{p}_{vj}. \tag{3.11}$$

Table 3.1: Joint specifications of Mitsubishi RV-2A robot arm

Joint	**Operating range (degree)**	**Speed of motion (degree/s)**
J1	320 (-160 to +160)	150
J2	180 (-135 to +45)	150
J3	120 (+50 to +170)	180
J4	320 (-160 to +160)	240
J5	240 (-120 to +120)	180
J6	400 (-200 to +200)	330

Similarly, the average orientation $\mathbf{R}_m$ can be derived from $\mathbf{R}_v$. In responding to the hand's moving speed in our experiments, we set $N = 200$ for the sampling frequency of 1 kHz, as a delay of 0.1 s, which led to stable maneuver. As for other parameters in the experiments, for matching up the workspaces between these two coordinates of the Phantom Omni and Mitsubishi RV-2A, the scaling factor k in Equation (3.1) was set to be 2. Values for the stiffness of the virtual tools and spring were empirically determined to be 0.8 N/mm and 0.2 N/mm, respectively, and α and f_{th} used in Algorithm 3 for the virtual bumper to be 0.01 mm/N and 50 N. To meet the requirements in computation, we use three threads to deal with the haptic data (1 kHz), visual data (60 Hz), and manipulator control (141 Hz), respectively.

3.4 Experiments

To evaluate its performance, we applied the proposed manipulation system to conduct two kinds of tasks: (i) executing linear movement with fixed orientation and (ii) manipulating the hex key for screw fastening. The first task is used to show that the manipulation system is able to help the user to yield accurate contour following on both position and orientation. The second one is to show its ability in dealing with contact during task execution. Five male and two female students (undergraduate or graduate in National Chiao Tung University, Taiwan), between 19 and 25 years old, all right-handed, were invited to perform the experiments. All of them were with no or little experience in manipulating the robot manipulator using the haptic input device before. To alleviate the learning effect, the order of the operating modes (illustrated below) was randomly arranged for each subject. They were asked to complete at least two successful trials for each operation mode. Before task execution, the subject practiced the manipulation several times to be familiar with the manipulation system.

3.4.1 Executing Linear Movement with Fixed Orientation

Fig. 3.9 shows the experimental setup, in which the user manipulated the Phantom Omni to move the hex key, mounted on the Mitsubishi RV-2A, along a straight path via points A, B, and C with fixed orientations. The procedure for movement is divided into three phases:

- Phase AB: start from point A, pick up the hex key, and then move it from point A to point B along a straight line, while maintaining its orientation.
- Transition phase: change its orientation to another one.
- Phase BC: start from point B, and move it from point B to point C with the current orientation.

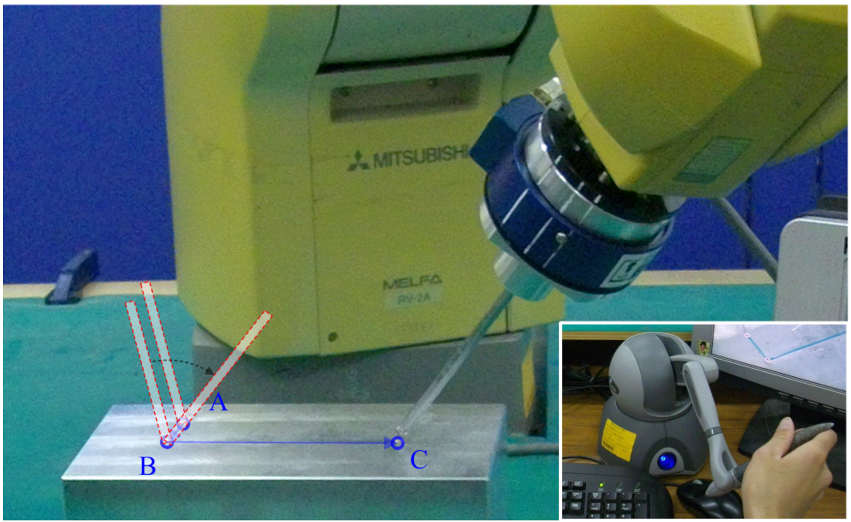
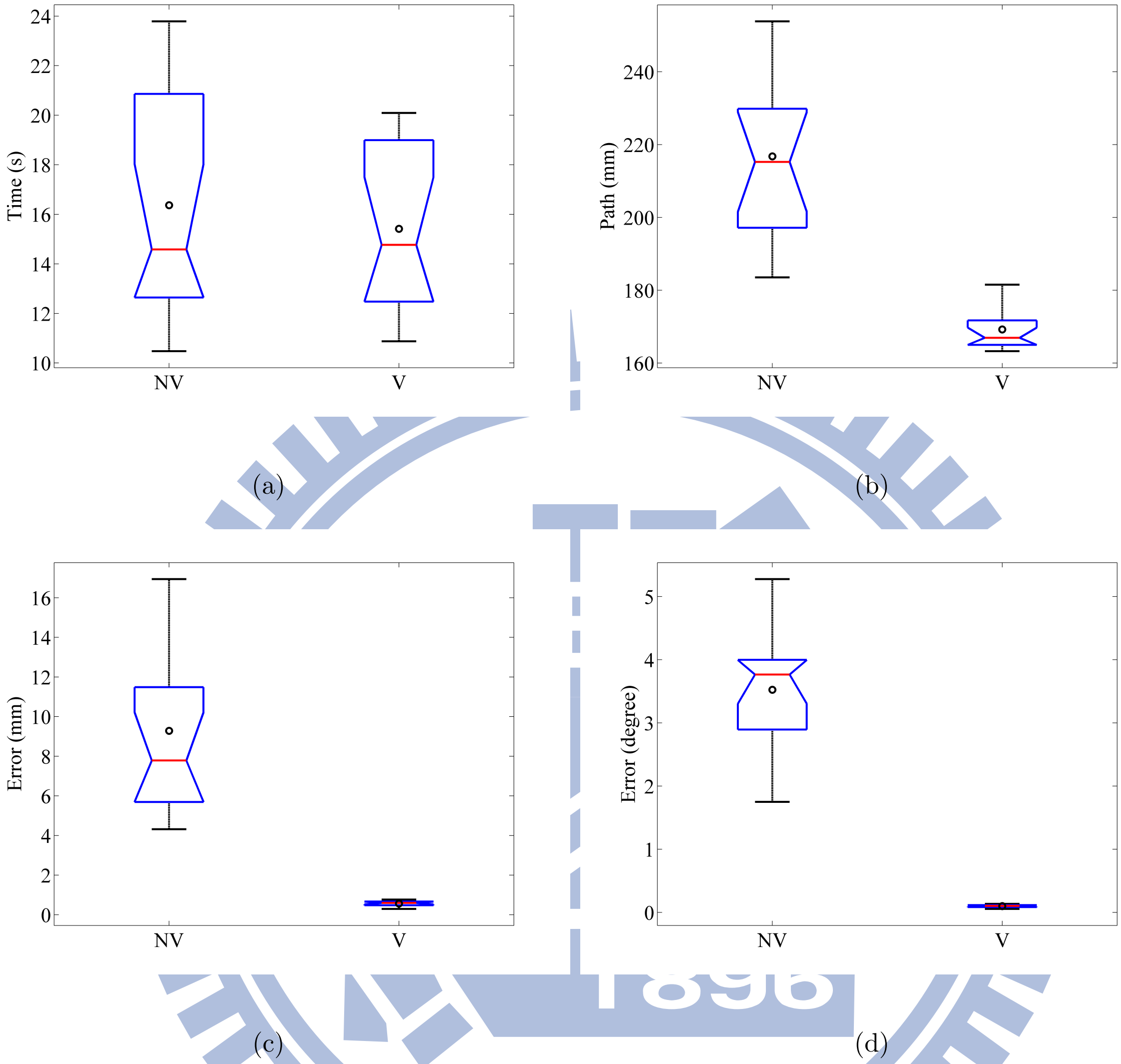


Fig. 3.9: Experimental setup for executing linear movement from point A to B, then C with fixed orientations.

In responding to these three phases, proper virtual tools are called for assistance in manipulation:

- Phases AB and BC: the line tool for assistance for the two forward-only straight motions with a stop point, and the N-rotation tool for maintaining the orientation.
- Transition phase: the point tool to let the bottom of the hex key be fixed at point B.

To compare the performance with and without the virtual tools, two operation modes were provided for the subject: NV (no virtual tools) and V (with virtual tools). Fig. 3.10 shows the experimental results in box plots. As Fig. 3.10(a) shows, the assistance of the virtual tools only slightly affected the execution time (the medians for mode NV: 14.6 s and mode V: 14.8 s) due to a short movement distance (15 cm from point A to C). In Fig. 3.10(b), the virtual tools did reduce the robot traveling path evidently (the medians



(a)

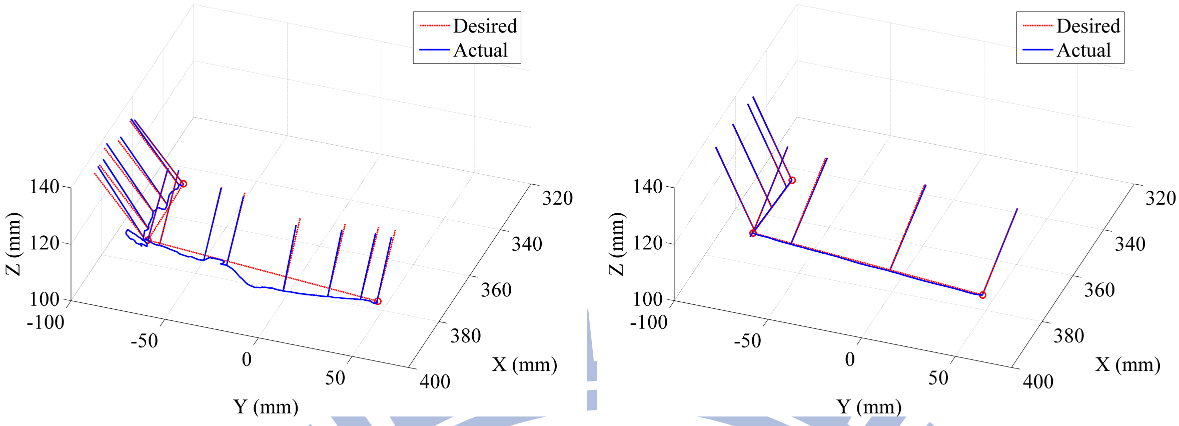
(b)

(c)

(d)

Fig. 3.10: Box plots for the linear movement task: (a) execution time, (b) robot traveling path, and (c)-(d) maximum position and orientation errors.

for mode NV: 215.3 mm and mode V: 166.9 mm, a 22.5% reduction). Figs. 3.10(c) and 3.10(d) show the maximum position and orientation errors during manipulation, respectively, in position: the medians for mode NV: 7.79 mm and mode V: 0.56 mm and in orientation: the medians for mode NV: 3.77°and mode V: 0.11°, which indicate the virtual tools significantly reduced these two errors. For further evaluation, Fig. 3.11 shows the desired and actual robot trajectories for user A as an example. In Fig. 3.11, with the assistance of the virtual tools, user A yielded a very straight trajectory while



(a) Mode NV

(b) Mode V

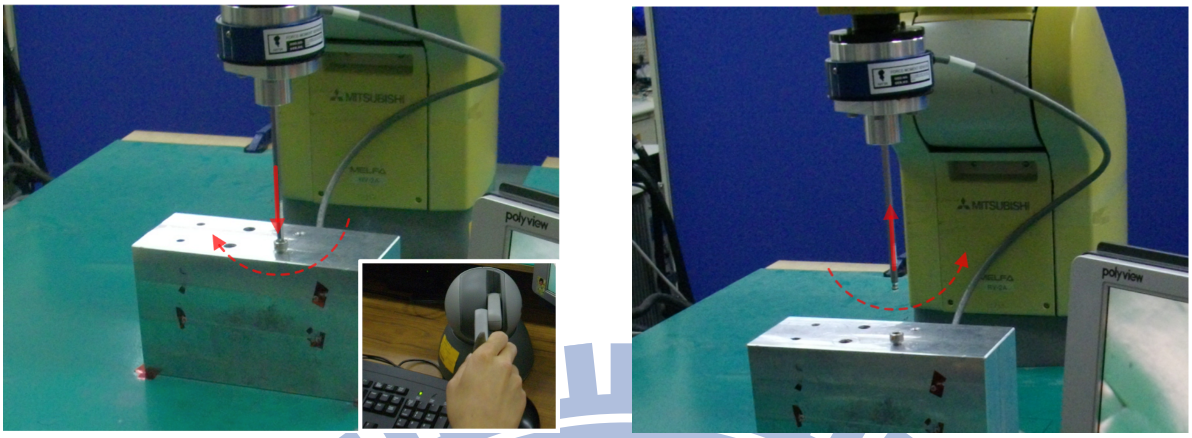
Fig. 3.11: Desired and actual robot trajectories for user A in (a) mode NV and (b) mode V.

maintaining the orientation of the hex key. The experimental results demonstrated that the virtual tools were very helpful in executing linear movement with fixed orientation.

3.4.2 Manipulating the Hex Key for Screw Fastening

Fig. 3.12 shows the experimental setup, in which the user manipulated the Phantom Omni to turn the hex key to fasten a hex socket screw. The procedure is as follows:

- Phase 1: place the ball end of the hex key into the hex socket screw in an upright orientation.
- Phase 2: rotate the hex key clockwise about a half turn to fasten the screw.
- Phase 3: release the hex key from the screw and then rotate it back counter-clockwise.
- Phase 4: repeat the steps above until the screw is fastened.



(a) (b)

Fig. 3.12: Experimental setup for manipulating the hex key for screw fastening: (a) place the ball end of the hex key into the hex socket screw and turn it clockwise to fasten the screw and (b) release the hex key from the screw by turning it counter-clockwise.

In responding to these operations, proper virtual tools are called for assistance in manipulation:

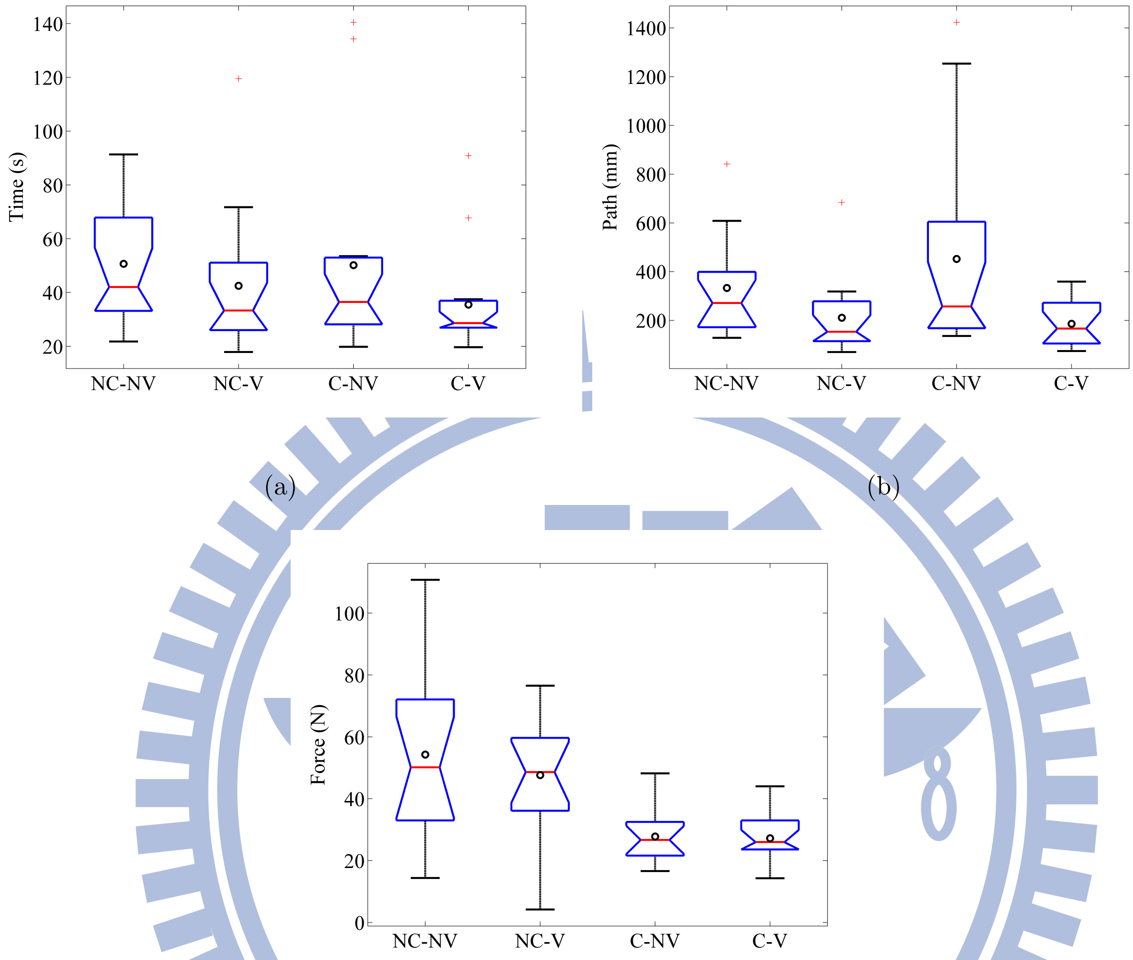
- Use the line tool to assist in the straight motion along with the rotating axis, which helps to place the hex key into the hex socket screw.
- Use the N-rotation tool to maintain a fixed rotating axis, which avoids the deviation due to unsteady hand movement.

In addition, we also provided the options of a combination of virtual bumper and spring, which was intended for contact protection, leading to totally four operation modes: NC-NV (no contact protection, no virtual tools), NC-V (no contact protection, with virtual tools), C-NV (with contact protection, no virtual tools), and C-V (with contact protection, with virtual tools). Table 3.2 lists the numbers of failures corresponding to the four operation modes. In Table 3.2, modes C-NV and C-V led to smaller numbers of

Table 3.2: Number of failures for the four operation modes

	Operation mode			
User	**NC-NV**	**NC-V**	**C-NV**	**C-V**
A	1	0	0	0
B	3	1	0	0
C	2	0	0	0
D	0	0	0	0
E	0	3	0	1
F	0	1	0	0
G	0	3	0	0
Sum	6 (40 %)	8 (53 %)	0 (0 %)	1 (7 %)

failures, indicating that the virtual bumper and spring did protect the robot manipulator from hard contact. Fig. 3.13 shows the box plots on execution time, traveling path, and average contact force. In Fig. 3.13(a), the virtual tools reduced the completion time up to about 8.8 s (20.9%) for mode NC-V vs. NC-NV and 7.9 s (21.6%) for mode C-V vs. C-NV, and in Fig. 3.13(b), they reduced the traveling path for about 117.8 mm (43.4%) for mode NC-V vs. NC-NV and 90.6 mm (35.2%) for mode C-V vs. C-NV. These results demonstrate that the virtual tools were quite helpful in performing the screw fastening task. As for contact force, Fig. 3.13(c) shows the virtual bumper and spring reduced the average contact force for about 23.5 N (46.8%) for mode C-NV vs. NC-NV and 22.6 N (46.5%) for mode C-V vs. NC-V. We further looked into the entire trajectories for the four modes during task execution. Fig. 3.14 shows the trajectories for user D as an example. With the help of the proposed virtual tools, the trajectories in Figs. 3.14(b)



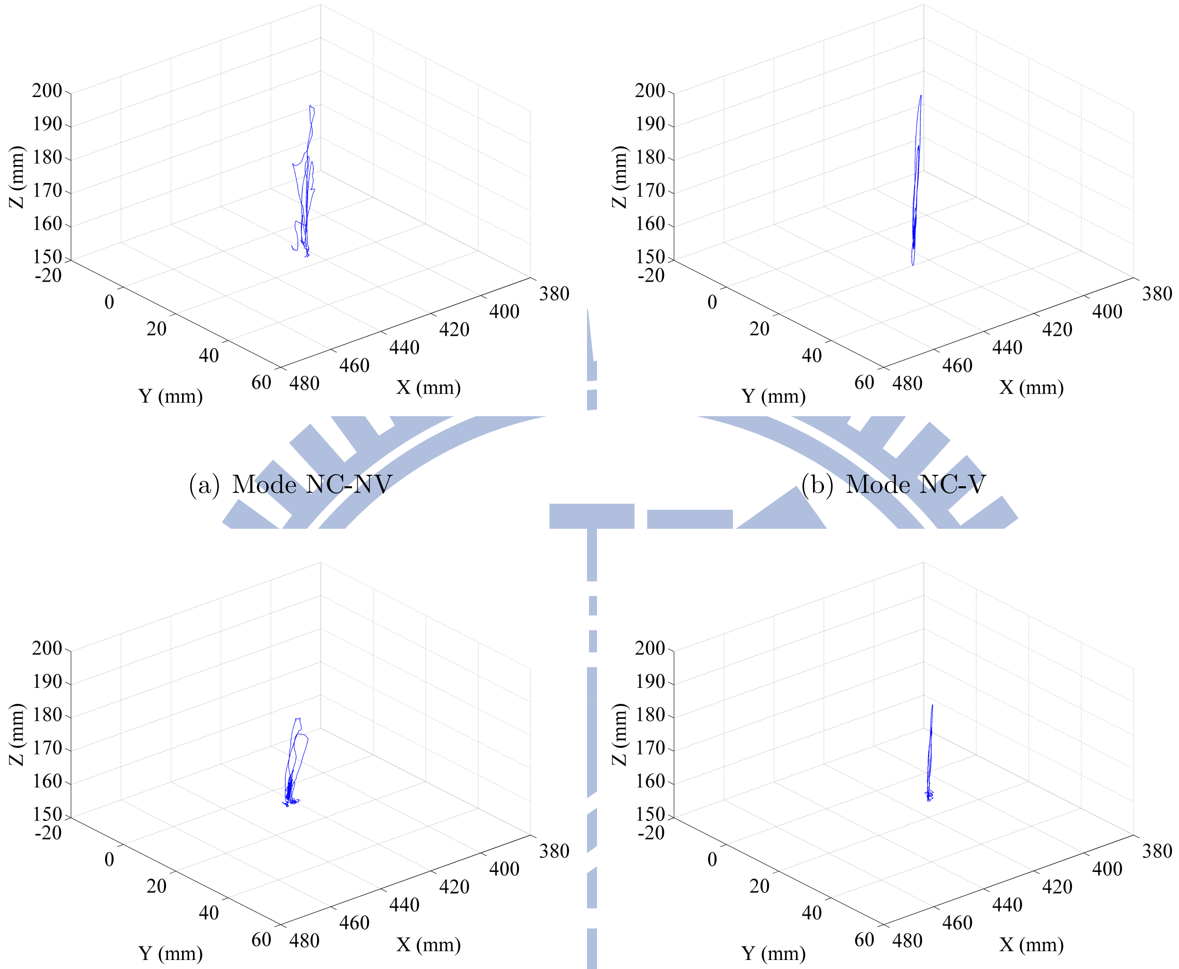
(a)

(b)

(c)

Fig. 3.13: Box plots for the screw-fastening task: (a) completion time, (b) traveling path, and (c) average contact force.

and 3.14(d) were straight and close together, compared with those in Figs. 3.14(a) and 3.14(c). We also observed how the virtual bumper and spring assisted in reducing the contact force during the process. Fig. 3.15 shows the trajectories of contact force and spring force for user D in the direction of Z axis, which is the main direction for contact. With their assistance, the contact forces were much reduced (the maximum force was reduced from over 400 N to below 200 N). Mode C-NV in Fig 3.15(a) and mode C-V in Fig 3.15(b) also show that the shapes of the spring forces were similar to those of the



(a) Mode NC-NV

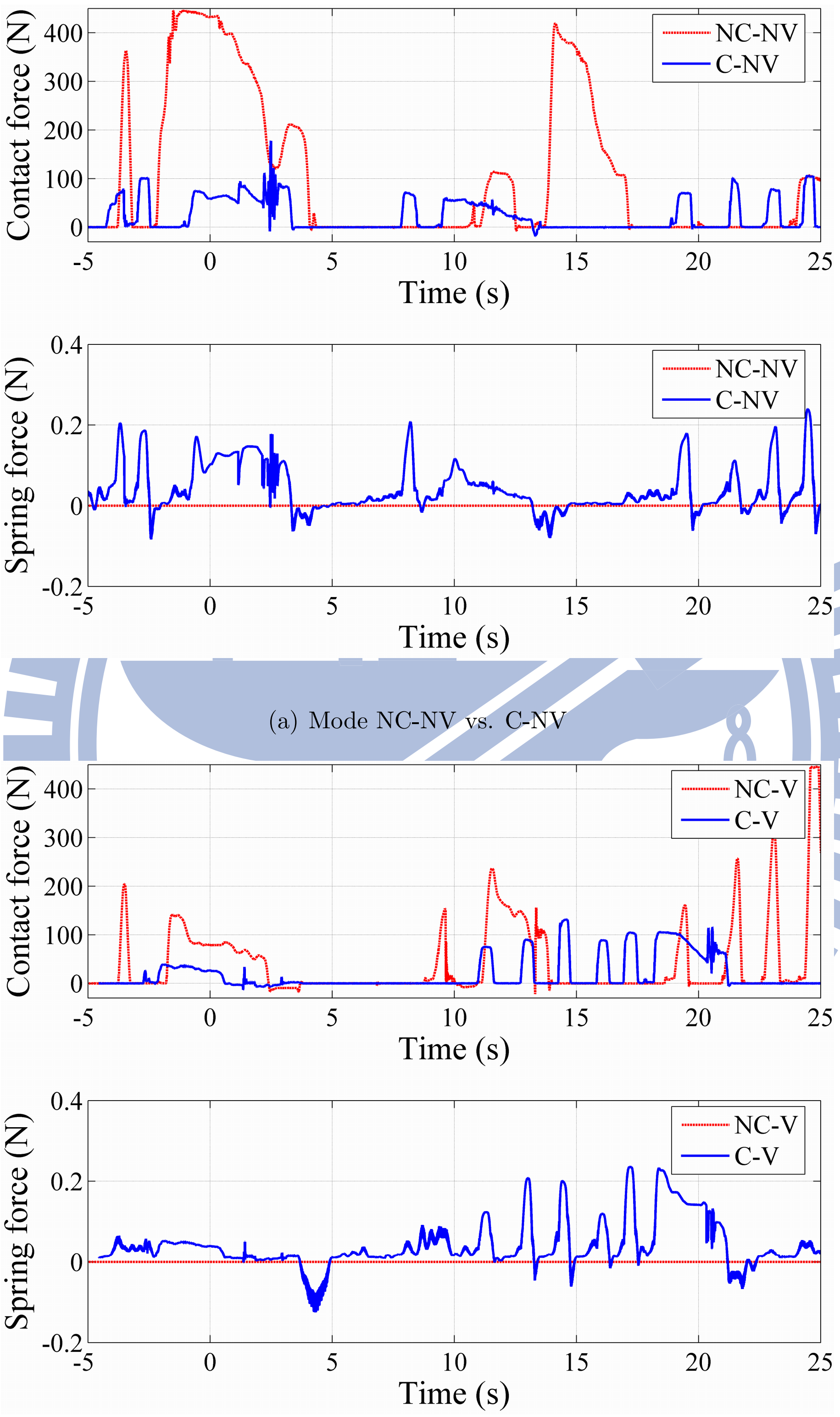
(b) Mode NC-V

(c) Mode C-NV

(d) Mode C-V

Fig. 3.14: Robot trajectories for user D in (a) mode NC-NV, (b) mode NC-V, (c) mode C-NV, and (d) mode C-V.

contact forces, indicating the virtual spring, which executed the command adjusted by the virtual bumper, did respond to the contact. From the interviews after the experiments, all the subjects considered mode C-V was the easiest one for task execution, and mode NC-NV most difficult. They also reported that the virtual tools were quite helpful for them to place the hex key into the screw socket and also to turn the screw, while the virtual bumper and spring kept them from worrying about collision.



(a) Mode NC-NV vs. C-NV

(b) Mode NC-V vs. C-V

Fig. 3.15: Trajectories of contact force and spring force for user D (Z-axis) in (a) mode NC-NV vs. C-NV and (b) mode NC-V vs. C-V.

3.5 Summary

In this chapter, we have proposed an effective manipulation system to assist the user for governing multi-DOF position-controlled robot manipulators. We have thus designed a set of virtual tools, spring, and bumper. During the experiments, the proposed manipulation system has been used to assist the execution of the contour following and screw fastening tasks. Satisfactory experimental results have demonstrated its effectiveness for these seven users with had no or little experience in manipulating the robot manipulator before.

Chapter 4 Conclusion and Future Work

This dissertation has presented two virtual motion constraint design methods for human-machine systems, which can be applicable to well-known or uncertain environments, and demonstrated their possible applications in simulation systems and robot manipulation.

For simulation systems, the proposed pixel-based method has been used to easily construct the desired motion constraints with various shapes via a graphic editing software, which can also achieve smooth force rendering between the transition of different virtual walls. We have also developed a multi-functional manipulation system based on a 2-DOF force-reflection joystick, which is designed to be versatile, but not for exact replication of the real mechanism. The emulations of an omni-directional wrench and a manual gearshift lever have shown certain degree of resemblance and reality. The proposed pixel-based method is ready for 3D force rendering as its implementation is in the pixel space, and can be easily extended to 3D cases, which are represented with voxels (volumetric pixel). However, it will induce more complexity for constructing 3D motion constraints in 3D space, compared with that in 2D cases. Therefore, in future works, we plan to develop a 3D modeling method for the proposed pixel-based method (for instance, transferring a 2D map into 3D map for haptic exploration, like [42]), or to generate the 3D representation of objects in the pixel space by utilizing the existing 3D modeling methods (e.g., 3D CAD models [38]). In addition, we also plan to introduce the temporal factor into motion constraint, making it vary along with task progress in applications, like a haptic recorder and player system for motor skill learning [16,41].

For manipulating a multi-DOF position-controlled robot manipulator without prior modeling environments, we have designed several virtual tools (point, line, plane, N-rotation, and F-rotation), which can be used to provide the position and orientation assistance to the user, according to her/his decision on site, to improve the human performance in manipulation in 3D space. In addition, we have also developed the virtual spring to establish the haptic linkage between the user and the end-effector and the virtual bumper to tackle with contact tasks. Based on a 6-DOF force-reflection joystick, the proposed manipulation system has been used to assist the execution of contour following and screw fastening. Satisfactory experimental results have shown its effectiveness on both accuracy and speed. In future works, to make the proposed system more friendly and easier to use, we plan to introduce a speech recognition system to help the user to call for various types of virtual tools. We also plan to integrate this system with more sensors, such as a Kinect [1] or a laser range finder, to design motion constraints, which can deal with varying environments in real-time, such as tracing a dynamic target. In addition, we will induce a torque-feedback joystick for studying the effects on orientation assistance with physical torque.

In conclusion, this dissertation has shown the applications of haptic devices, based on the concept of virtual motion constraint. To further verify the effectiveness of these manipulative systems, which can be also applied to a wider range of users, inviting more people with different ages or genders to perform these experiments is demanded. In the near future, we can expect that haptic devices will become cheaper, lighter, and more powerful, and play a more important role in human-machine systems, which can much enhance human-machine interaction.

Bibliography

[1] (2012, March) Kinect. Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/Kinect

[2] (2012, March) Phantom omni haptic device. Sensable Technologies. [Online]. Available: http://www.sensable.com/haptic-phantom-omni.htm

[3] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 897–903.

[4] J. Abbott, P. Marayong, and A. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, S. Thrun, R. Brooks, and H. Durrant-Whyte, Eds. Springer Berlin/Heidelberg, 2007, vol. 28, pp. 49–64.

[5] J. J. Abbott and A. M. Okamura, "Pseudo-admittance bilateral telemanipulation with guidance virtual fixtures," *Robotics Research*, vol. 26, no. 8, pp. 865–884, 2007.

[6] J. Abbott, G. Hager, and A. Okamura, "Steady-hand teleoperation with virtual fixtures," in *IEEE International Workshop on Robot and Human Interactive Communication*, 2003, pp. 145–151.

[7] Y. Adachi, T. Kumano, and K. Ogino, "Intermediate representation for stiff virtual objects," in *IEEE Virtual Reality Annual International Symposium*, 1995, pp. 203–211.

[8] A. Bettini, P. Marayong, S. Lang, A. Okamura, and G. Hager, “Vision-assisted control for manipulation using virtual fixtures,” *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 953–966, 2004.

[9] G. C. Burdea, *Force and Touch Feedback for Virtual Reality.* New York: John Wiley & Sons, 1996.

[10] G. C. Burdea, “Invited review: the synergy between virtual reality and robotics,” *IEEE Transaction on Robotics and Automation*, vol. 15, no. 3, pp. 400–410, 1999.

[11] G. C. Burdea and P. Coiffet, *Virtual Reality Technology.* New York: John Wiley & Sons, 1994.

[12] C.-C. P. Chu, T. H. Dani, and R. Gadh, “Multimodal interface for a virtual reality based computer aided designed system,” in *IEEE International Conference on Robotics and Automation*, 1997, pp. 1329–1334.

[13] C. L. Clover, G. R. Luecke, J. J. Troy, and W. A. McNeely, “Dynamic simulation of virtual mechanisms with haptic feedback using industrial robotics equipment,” in *IEEE International Conference on Robotics and Automation*, 1997, pp. 724–730.

[14] J. E. Colgate, P. E. Grafing, M. C. Stanley, and G. Schenkel, “Implementation of stiff virtual walls in force-reflecting interfaces,” in *IEEE Virtual Reality Annual International Symposium*, 1993, pp. 202–208.

[15] J. Colgate, M. Stanley, and J. Brown, “Issues in the haptic display of tool use,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995, pp. 140–145.

[16] M. Eid, M. Mansour, R. Iglesias, and A. El Saddik, "A device independent haptic player," in *IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, 2007, pp. 7–12.

[17] B. A. Forsyth and K. E. MacLean, "Predictive haptic guidance: Intelligent user assistance for the control of dynamic tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 1, pp. 103–113, 2006.

[18] A. Frisoli, C. A. Avizzano, and M. Bergamasco, "Simulation of a manual gearshift with a 2 DOF force-feedback joystick," in *IEEE International Conference on Robotics and Automation*, 2001, pp. 1364–1369.

[19] A. Gasparetto, R. Vidoni, and V. Zanotto, "Dforce: Delayed force reference control for master-slave robotic systems," *Mechatronics*, vol. 19, no. 5, pp. 639–646, 2009.

[20] T. L. Gibo, L. N. Verner, D. D. Yuh, and A. M. Okamura, "Design considerations and human-machine performance of moving virtual fixtures," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 671–676.

[21] C. Glover, B. Russell, A. White, M. Miller, and A. Stoytchev, "An effective and intuitive control interface for remote robot teleoperation with complete haptic feedback," in *Emerging Technologies Conference*, 2009.

[22] C. Guo and E. Sharlin, "Exploring the use of tangible user interfaces for human-robot interaction: a comparative study," in *SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 121–130.

[23] B. Hannaford, L. Wood, D. McAffee, and H. Zak, "Performance evaluation of a six-axis generalized force-reflecting teleoperator," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 3, pp. 620–633, 1991.

[24] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, and G. R. de-la Torre, "Haptic interfaces and devices," *Sensor review*, vol. 24, no. 1, pp. 16–29, 2004.

[25] M.-C. Hsieh, P.-Y. Tseng, M.-S. Jan, and K.-Y. Young, "Manipulating a multi-DOF robot manipulator for tasks in home-like environments," in *17th International Symposium on Artificial Life and Robotics*, 2012.

[26] M.-C. Hsieh and K.-Y. Young, "Motion constraint design and implementation for a multi-functional virtual manipulation system," *Mechatronics*, vol. 20, no. 3, pp. 346–354, 2010.

[27] J. Hyde and M. Cutkosky, "Contact transition control: an experimental study," in *IEEE International Conference on Robotics and Automation*, 1993, pp. 363–368.

[28] L. D. Joly and C. Andriot, "Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism," in *IEEE International Conference on Robotics and Automation*, 1995, pp. 357–362.

[29] C. C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 20–29, 2007.

[30] C.-P. Kuan and K.-Y. Young, "VR-based teleoperation for robot compliance control," *Journal of Intelligent and Robotic Systems*, vol. 30, no. 4, pp. 377–398, 2001.

[31] A. B. Kuang, S. Payandeh, B. Zheng, F. Henigman, and C. L. MacKenzie, "Assembling virtual fixtures for guidance in training environments," in *International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2004, pp. 367–374.

[32] K. J. Kuchenbecker and G. Niemeyer, "Induced master motion in force-reflecting teleoperation," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 4, pp. 800–810, 2006.

[33] S. Lederman and R. Klatzky, "Haptic perception: A tutorial," *Attention, Perception, & Psychophysics*, vol. 71, pp. 1439–1459, 2009.

[34] M. Li, M. Ishii, and R. H. Taylor, "Spatial motion constraints using virtual fixtures generated by anatomy," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 4–19, 2007.

[35] M. Li and A. M. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in *International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003, pp. 125–131.

[36] C.-W. Liang, M.-C. Hsieh, and K.-Y. Young, "A VR-based calligraphy writing system with force reflection," *International Journal of Automation and Smart Technology*, vol. 1, no. 2, pp. 83–91, 2011.

[37] W.-C. Lin and K.-Y. Young, "Design of force-reflection joystick system for VR-based simulation," *Journal of Information Science and Engineering*, vol. 23, no. 5, pp. 1421–1436, 2007.

[38] X. Liu, G. Dodds, J. McCartney, and B. Hinds, "Virtual DesignWorks-designing 3D CAD models via haptic interaction," *Computer-Aided Design*, vol. 36, no. 12, pp. 1129–1140, 2004.

[39] M. Minsky, M. Ouh-young, O. Steele, F. P. B. Jr., and M. Behensky, "Feeling and seeing: Issues in force display," *Computer Graphics*, vol. 24, no. 2, pp. 235–243, 1990.

[40] *Industrial Robots Instruction Manual, Ethernet Interface CRn-500 series*, Mitsubishi Electric Industrial Automation, 2002.

[41] D. Morris, H. Tan, F. Barbagli, T. Chang, and K. Salisbury, "Haptic feedback enhances force skill learning," in *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2007, pp. 21–26.

[42] K. Moustakas, G. Nikolakis, K. Kostopoulos, D. Tzovaras, and M. Strintzis, "Haptic rendering of visual data for the visually impaired," *IEEE Multimedia*, vol. 14, no. 1, pp. 62–72, 2007.

[43] A. Nahvi, D. D. Nelson, J. M. Hollerbach, and D. E. Johnson, "Haptic manipulation of virtual mechanisms from mechanical CAD designs," in *IEEE International Conference on Robotics and Automation*, 1998, pp. 375–380.

[44] H. Ohtsuka, K. Shibasato, and S. Kawaji, "Experimental study of collaborater in human-machine system," *Mechatronics*, vol. 19, no. 4, pp. 450–456, 2009.

[45] M. Ouh-young, W.-N. Tsai, M.-C. Tsai, J.-R. Wu, C.-H. Huang, and T.-J. Yang, "Low-cost force feedback joystick and its use in PC video games," *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 787–794, 1995.

[46] H. Park, E. Kim, S. Jang, S. Park, M. Park, and H. Kim, “HMM-based gesture recognition for robot control,” in *Pattern Recognition and Image Analysis*, ser. Lecture Notes in Computer Science, J. Marques, N. Prez de la Blanca, and P. Pina, Eds. Springer Berlin/Heidelberg, 2005, vol. 3522, pp. 607–614.

[47] C. Passenberg, A. Peer, and M. Buss, “A survey of environment-, operator-, and task-adapted controllers for teleoperation systems,” *Mechatronics*, vol. 20, no. 7, pp. 787–801, 2010.

[48] M. A. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. A. Moore, R. B. Gillespie, and P. Akella, “Cobot architecture,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, 2001.

[49] R. Prada and S. Payandeh, “A study on design and analysis of virtual fixtures for cutting in training environments,” in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2005, pp. 375–380.

[50] R. Prada and S. Payandeh, “On study of design and implementation of virtual fixtures,” *Virtual Reality*, vol. 13, no. 2, pp. 117–129, 2009.

[51] P. Rizun and G. Sutherland, “Tactile feedback laser system with applications to robotic surgery,” in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2005, pp. 426–431.

[52] G. Robles-De-La-Torre, “The importance of the sense of touch in virtual and real environments,” *IEEE Multimedia*, vol. 13, no. 3, pp. 24–30, 2006.

[53] L. B. Rosenberg, “Virtual fixtures: Perceptual tools for telerobotic manipulation,” in *IEEE Virtual Reality Annual International Symposium*, 1993, pp. 76–82.

[54] D. C. Ruspini, K. Kolarov, and O. Khatib, “The haptic display of complex graphical environments,” in *ACM International Conference on Computer Graphics and Interactive Techniques*, 1997, pp. 345–352.

[55] D. Ryu, J.-B. Song, C. Cho, S. Kang, and M. Kim, “Development of a six DOF haptic master for teleoperation of a mobile manipulator,” *Mechatronics*, vol. 20, no. 2, pp. 181–191, 2010.

[56] K. Salisbury, F. Conti, and F. Barbagli, “Haptic rendering: Introductory concepts,” *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 24–32, 2004.

[57] M. A. Srinivasan and C. Basdogan, “Haptics in virtual environments: Taxonomy, research status, and challenges,” *Computer & Graphics*, vol. 21, no. 4, pp. 393–404, 1997.

[58] B. Stanczyk, A. Peer, and M. Buss, “Development of a high-performance haptic telemanipulation system with dissimilar kinematics,” *Advanced Robotics*, vol. 20, no. 11, pp. 1303–1320, 2006.

[59] A. R. Tilley, *The Measure of Man and Woman — Human Factors in Design*. New York: John Wiley & Sons, 2002.

[60] E. Veras, K. Khokar, R. Alqasemi, and R. Dubey, “Laser-assisted real-time and scaled telerobotic control of a manipulator for defense and security applications,” in *SPIE Defense, Security, and Sensing Conference*, vol. 7332, no. 1, 2009.

[61] D. Wang, Y. Zhang, and C. Yao, "Stroke-based modeling and haptic skill display for Chinese calligraphy simulation system," *Virtual Reality*, vol. 9, pp. 118–132, 2006.

[62] Y. Wang, Y. Chen, Z. Nan, and Y. Hu, "Study on welder training by means of haptic guidance and virtual reality for arc welding," in *IEEE International Conference on Robotics and Biomimetics*, 2006, pp. 954–958.

[63] W. Yu, R. Alqasemi, R. Dubey, and N. Pernalete, "Telemanipulation assistance based on motion intention recognition," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 1121–1126.

[64] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *IEEE International Conference on Intelligent Robots and Systems*, 1995, pp. 146–151.

Vita

Mu-Cheng Hsieh was born in Miaoli, Taiwan, 1969. In 1989, he graduated from the five-year program in the Division of Electrical Engineering, National Taipei Institute of Technology (renamed to National Taipei University of Technology in 1997). He received the M.S. degree in Electrical and Control Engineering from National Chiao Tung University in 2001. Since 1993, he has been working at National Tsing Hua University, where he is currently a Technical Specialist at the Computer and Communication Center. His research interests include human-machine systems, Internet applications, computer haptics, robot control, and virtual motion constraint design.

Publication list

Journal papers

- M.-C. Hsieh and K.-Y. Young, "Motion constraint design and implementation for a multi-functional virtual manipulation system," *Mechatronics*, vol. 20, no. 3, pp. 346–354, 2010.
- C.-W. Liang, M.-C. Hsieh, and K.-Y. Young, "A VR-based calligraphy writing system with force reflection," *International Journal of Automation and Smart Technology*, vol. 1, no. 2, pp. 83–91, 2011.
- M.-C. Hsieh and K.-Y. Young, "Effective manipulation for a multi-DOF robot manipulator in laboratory environments," accepted by *Journal of the Chinese Institute of Engineers*, 2012.

- M.-C. Hsieh and K.-Y. Young, “Manipulating a multi-DOF robot manipulator via virtual tools,” submitted to *Journal of Intelligent and Robotic Systems.*

Conference papers

- M.-C. Hsieh and K.-Y. Young, “Motion constraint design and implementation for a multi-functional virtual manipulation system,” *CACS Automatic Control Conference*, pp. 304–309, 2006.
- M.-C. Hsieh and K.-Y. Young, “Motion constraint design for a multi-functional virtual manipulation system,” *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1579–1584, 2008.
- M.-C. Hsieh, P.-Y. Tseng, M.-S. Jan, and K.-Y. Young, “Manipulating a multi-DOF robot manipulator for tasks in home-like environments,” *International Symposium on Artificial Life and Robotics*, 2012.