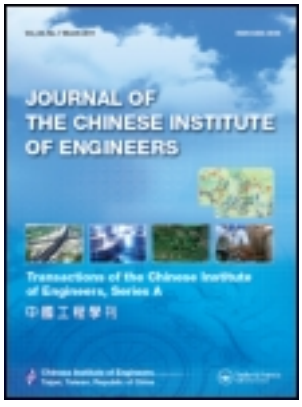

Journal of the Chinese Institute of Engineers
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcie20

Effective manipulation for a multi-DOF robot manipulator in laboratory environments

Mu-Cheng Hsieh [a] & Kuu-Young Young [a]
[a] Department of Electrical Engineering , National Chiao-Tung University Vision Research Center, National Chiao-Tung University , 1001 University Road, Hsinchu , Taiwan 300 , ROC
Published online: 13 Nov 2012.

To cite this article: Mu-Cheng Hsieh & Kuu-Young Young (2013) Effective manipulation for a multi-DOF robot manipulator in laboratory environments, Journal of the Chinese Institute of Engineers, 36:5, 566-576, DOI: 10.1080/02533839.2012.737112

To link to this article: http://dx.doi.org/10.1080/02533839.2012.737112

Journal of the Chinese Institute of Engineers, 2013
Vol. 36, No. 5, 566–576, http://dx.doi.org/10.1080/02533839.2012.737112

Effective manipulation for a multi-DOF robot manipulator in laboratory environments

Mu-Cheng Hsieh and Kuu-Young Young*

Department of Electrical Engineering, National Chiao-Tung University Vision Research Center, National Chiao-Tung University, 1001 University Road, Hsinchu, Taiwan 300, ROC

(*Received 6 May 2011; final version received 9 February 2012*)

Robots are expected to enter human environments soon, and many challenging problems may then emerge when facing uncertain and varying environments. Among the challenges, one issue of great interest is how they can be effectively operated for task execution. If a robot is not to be controlled by detailed analysis and program coding, one appealing alternative is to provide a kind of manipulation system which enables the user to operate the robot naturally and efficiently. This idea has motivated us to develop an effective manipulation system for multi-degree of freedom (DOF) robot manipulators based on a 6-DOF haptic device. In this article, we focus on the laboratory environment, which is somewhat organized, but still demands sophisticated human manipulation. The proposed manipulation system, aimed for 3-D applications, provides two kinds of assistance. One is to enhance the linkage between the user and robot manipulator via the haptic clue. The other is to guide the movement of the user during task execution via a set of virtual tools, e.g., a ruler. We also propose a method for achieving smooth force rendering in manipulation, especially during the transition between two consecutive manipulations for guidance. For performance evaluation, this system is employed to conduct experiments in the chemical laboratory, which involve delicate and challenging maneuvers.

Keywords: manipulation system; haptic device; virtual tool; laboratory environment

1. Introduction

Due to rapid progress in sensing, control, and decision-making techniques, robot manipulators nowadays are able to handle tasks of high complexity. In the near future, more robots will enter our offices, laboratories, and families, conducting experiments, providing family and personal assistance, and other uses. Consequently, in dealing with uncertain and varying environments, rather than the organized environments in factories, many challenging problems shall emerge (Kemp *et al.* 2007, Lai *et al.* 2007). To date, robots still demand human's assistance when working in human environments. To enhance their usefulness, one issue of much interest is how we can effectively operate them for task execution. If we are not going to take on the load of detailed task analysis and program coding, one alternative is to let the user manipulate the robot via a proper means directly. In other words, an effective manipulation system should be provided for the user to achieve natural and efficient robot control. As traditional manipulative devices, like teach box, mouse, keyboard, and joystick, may not satisfy the requirements, we propose developing a dexterous manipulation system for multi- degree of freedom (DOF) robot manipulators based on a 6-DOF haptic device. Before tackling general human environments, in this study, we focus on the laboratory environment, which is somewhat organized, but still complicated for automated execution.

Among previous research works on manipulation in human environments, several intuitive manipulative devices have been proposed, such as camera-based devices (Park *et al.* 2005) and tangible user interfaces, like Wiimote (Guo and Sharlin 2008), which recognize user's gestures for robot governing. They usually do not provide intuitive force feedback though. Some have proposed using an identical robot as the user interface, but that comes at a high cost (Glover *et al.* 2009). For the proposed manipulation system, we adopt the 6-DOF force-reflection joystick as the manipulative device, which supports mutual interaction, involves both position and force information, and has the merits of simplicity and generality (Nahvi *et al.* 1998, Kuan and Young 2001, Preusche *et al.* 2002, Song *et al.* 2006, Stanczyk *et al.* 2006, He and Chen 2008). With this manipulation system, we design virtual motion constraints, which are based on the concept of virtual mechanisms and virtual fixtures

*Corresponding author. Email: kyoung@mail.nctu.edu.tw

previously proposed (Rosenberg 1993, Joly and Andriot 1995, Peshkin *et al.* 2001, Marayong *et al.* 2003, Abbott *et al.* 2007, He and Chen 2009, Prada and Payandeh 2009, Veras *et al.* 2009), to confine the movement of the user for 3-D tasks, according to the status of the robot manipulator and task progress. In other words, the virtual motion constraints are designed (i) to help the user recognize the spatial deviation between her/his desired position and the end-effector via a haptic clue and (ii) to guide her/his movement according to task requirements via some virtual tools. For instance, a virtual ruler can be used to help the user to move the robot along a straight line fast and precisely. The proposed manipulation system thus provides two kinds of virtual motion constraints to assist the user in an interactive and real-time manner. First, a virtual spring is installed between the user and robot manipulator to enhance a virtual linkage between them. In addition, a 3-D graphical environment is also implemented to foster the virtual linkage in a visual way. Second, three basic types of virtual tools (point, line, and plane) are furnished according to the user's demand on site, but not in a predefined environment or predicted way (Li and Okamura 2003, Kuang *et al.* 2004, Yu *et al.* 2004, Prada and Payandeh 2005, Forsyth and MacLean 2006). Meanwhile, we also propose a method for smooth force rendering during the transition between the uses of two consecutive tools in guidance. To demonstrate its effectiveness, the manipulation system is utilized to govern the robot manipulator to conduct the experiments in the chemical laboratory. During the experiments, several liquid pouring and tube moving actions occur, which demand very delicate maneuvers from the experimenter. Users of various backgrounds are invited for the experiments, and their responses analyzed. The rest of this article is organized as: Section 2 describes the proposed manipulation system and how the virtual linkage and virtual tools are constructed. System implementation is presented in Section 3. Section 4 discusses the experiments and performance evaluation. Finally, conclusions are given in Section 5.

2. Proposed manipulation system

Figure 1(a) illustrates the two major goals of the proposed manipulation system for a multi-DOF manipulator: (i) letting the user feel like she/he is

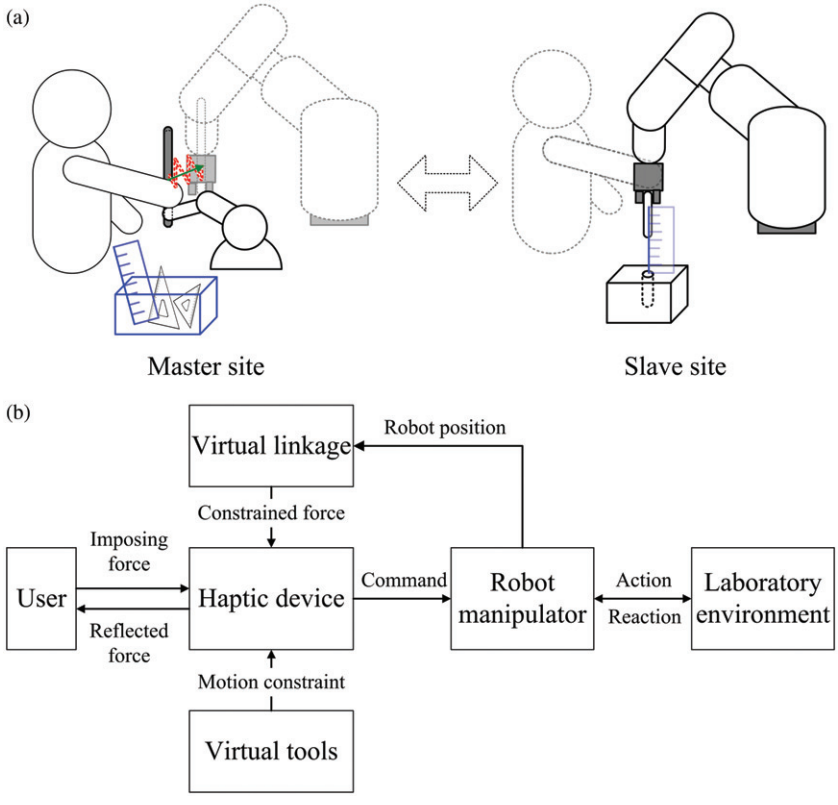

Figure 1. The proposed manipulation system for a multi-DOF manipulator: (a) conceptual diagram and (b) system block diagram.

manipulating the robot on site directly when teleoperating the task and (ii) providing a set of virtual tools that guide the hand movement of the user. Its system block diagram is shown in Figure 1(b), which consists of a 6-DOF haptic device, a virtual linkage, and virtual tools. Via the haptic device, the user sends in the commands to govern the robot manipulator at the slave site to perform the task in the laboratory environment. Using feedback, the robot position, via the virtual linkage and then the haptic device, becomes the reflected force, which is sent back to the user for following manipulation. Meanwhile, the system utilizes the virtual tools to furnish motion constraints for guidance. Based on the descriptions above, the success of the proposed manipulation system mainly depends on proper design of the virtual linkage and virtual tools. The former establishes the human–robot connection for natural and effective manipulation, and the latter provides proper assisting motion constraints along with the progress of task execution. In addition, because the maximum force of most haptic devices is smaller than that of a human, we assume that the human user is aware that the virtual motion constraint serves as the reminder and guidance, and is expected not to push hard on them intentionally. In Section 2.1, we describe how to develop the virtual linkage. In Section 2.2, we discuss how to implement the virtual tools for various types of tasks.

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. Figure 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 immersed 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 3-D graphical system to build the virtual robot manipulator to be much like the real one in both appearance and motion. For the haptic counterpart, we intend to let the user feel like her/his hand is holding 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 Figure 3. We intend to find a transformation that maps position $\boldsymbol{P}_M$ and orientation $\boldsymbol{R}_M$ in the coordinate system of the joystick (centered at $\boldsymbol{C}_M$) into the

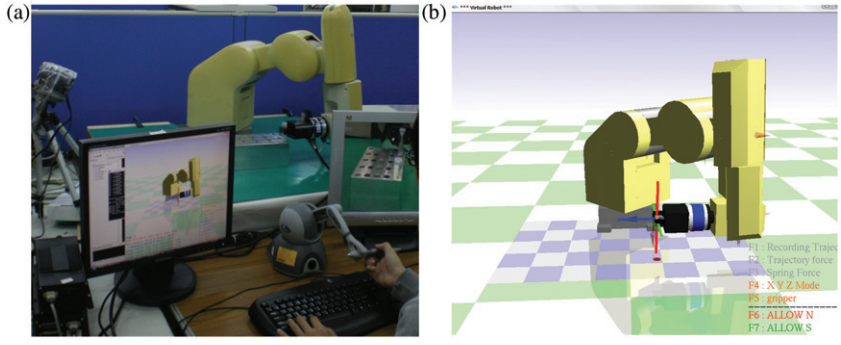

Figure 2. The Proposed manipulation system to manipulate: (a) the real robot manipulator via (b) the virtual robot manipulator.

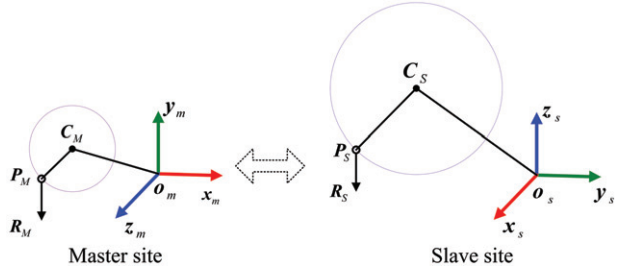

Figure 3. Alignment and workspace mapping between the coordinate systems of the joystick and robot manipulator.

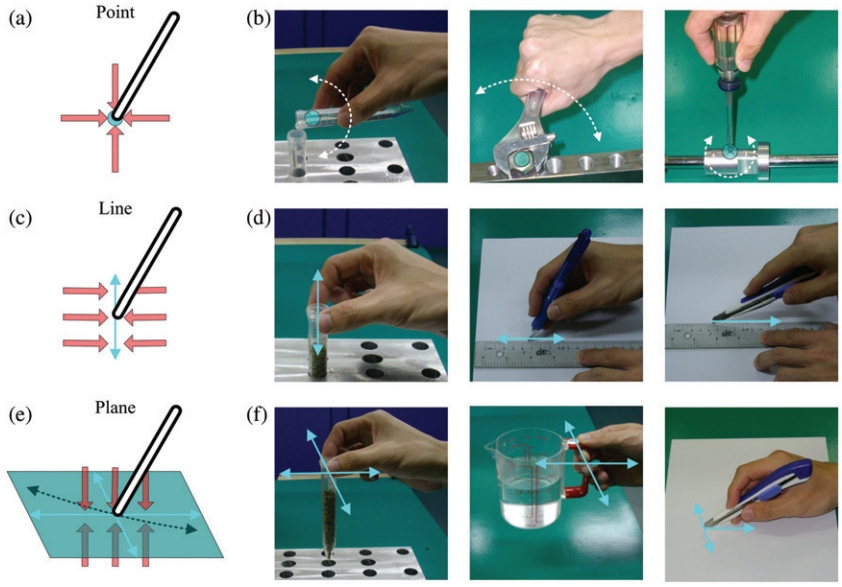

Figure 4. Point, line, and plane virtual tools and their possible applications: (a)–(b) point, (c)–(d) line, and (e)–(f) plane.

corresponding position $\boldsymbol{P}_S$ and orientation $\boldsymbol{R}_S$ in that of the robot manipulator (centered at $\boldsymbol{C}_S$), with k as the scaling factor for matching up the size of the two workspaces. The transformations are formulated as

$$\boldsymbol{P}_S = k\boldsymbol{R}_m^s(\boldsymbol{P}_M - \boldsymbol{C}_M) + \boldsymbol{C}_S, \tag{1a}$$

$$\boldsymbol{R}_S = \boldsymbol{R}_m^s\boldsymbol{R}_M, \tag{1b}$$

where $\boldsymbol{R}_m^s$ is the rotational matrix from the coordinate system of the joystick to that of the robot manipulator.

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 $\boldsymbol{F}_s$ is formulated as

$$\boldsymbol{F}_s = \boldsymbol{K}_s(\boldsymbol{P}_r - \boldsymbol{P}_h), \tag{2}$$

where K_s stands for the stiffness of the virtual spring and P_r and P_h the position of the end-effector and force-reflection joystick, respectively, both with respect to the coordinate system of the master.

2.2. *Virtual tools*

To tackle common operations in laboratory environments, we design three types of virtual tools: point, line, and plane. Figure 4 shows these three virtual tools and their possible applications. In Figure 4(a), the point tool confines the haptic device to rotate around a fixed point, so that it can be applied for tasks, such as liquid pouring and wrench or screw driver turning, as shown in Figure 4(b). In Figure 4(c), the line tool confines the haptic device to move along a straight line, just like a ruler. Its applications include moving a tube into a hole, drawing a straight line, and cutting a piece of paper with a ruler, as shown in Figure 4(d). In Figure 4(e), the plane tool confines the haptic device to move within a plane, and the applications include moving the tube around the rack or a cup of water above a table and also 2-D drawing, as shown in Figure 4(f). Note that, in some applications, it may be helpful to constrain the direction of the movement to be only forward or backward, for instance, pulling a tube out from a hole or cutting a piece of paper. This directional constraint is also implemented in the virtual tools.

When the virtual tools are applied for manipulation assistance, they will be placed close to the tip of the virtual haptic device (simultaneously the end-effector of the virtual robot manipulator) in an appropriate orientation. For example, when the line tool is used as a ruler, it will be placed alongside the haptic device and aligned with the direction in which it is to move. The virtual tool will generate a resistant force, whenever the haptic device deviates from its constraint, thus confining the robot manipulator to follow the desired movement. We adopt a spring model to generate the constrained force $\boldsymbol{F}_n$, formulated

$$\boldsymbol{F}_n = \boldsymbol{K}(\boldsymbol{C}_n - \boldsymbol{P}_n), \tag{3}$$

where $\boldsymbol{P}_n$ stands for the tip position of the haptic device, K the stiffness of virtual tool for adjusting the constrained force, and $\boldsymbol{C}_n$ the nearest tool point (NTP)

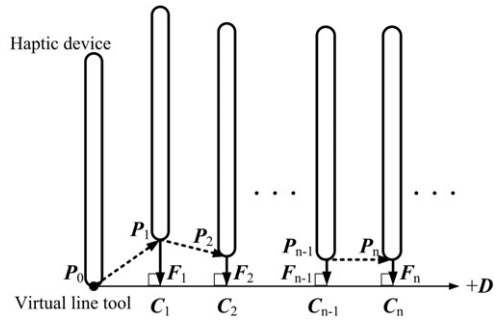

Figure 5. An example of applying the line tool for movement assistance.

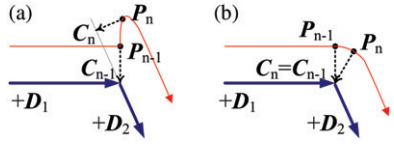

Figure 6. Two force rendering methods for the transition of two consecutive tools: (a) the orthogonal-projection method (unsmooth force rendering) and (b) the proposed method (smooth force rendering).

on the line tool that corresponds to $\boldsymbol{P}_n$ Figure 5 shows an example of how a line tool constrains the movement of the haptic device. In Figure 5, the line tool, orientated in a $+\boldsymbol{D}$ direction (later we denote such a forward-only line tool by $+\boldsymbol{D}$), is called upon to assist, at the moment the haptic device moves to point $\boldsymbol{P}_0$ Due to an imprecise maneuver by the user, the haptic device deviated at point $\boldsymbol{P}_1$ causing the line tool to generate a series of constrained forces, $\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots,$ to pull it back to the proximity of the line specified by the line tool.

Because a virtual tool traveling in a different direction, even an other tool (e.g., point or plane), may be demanded along with the progress of the task, it is not that straightforward to locate the NTPs for computing the corresponding constrained force as the case shown in Figure 5. As an example, Figure 6(a) shows the transition from tool $+\boldsymbol{D}_1$ at $\boldsymbol{P}_{n-1}$ to another $+\boldsymbol{D}_2$. If we only consider the orthogonal projection alone, $\boldsymbol{C}_n$ for $\boldsymbol{P}_n$ will be located at the location as shown in Figure 6(a) due to the presence of $+\boldsymbol{D}_2$, leading to unsmooth force rendering. To achieve smooth force rendering, our strategy is to keep tracking previous NTPs when locating the next one, just like god-object, proxy, and pixel-based algorithms do (Zilles and Salisbury 1995, Ruspini *et al.* 1997, Hsieh and Young 2010). Figure 6(b) shows how the proposed method works for the same case in Figure 6(a). In Figure 6(b), when the tool $+\boldsymbol{D}_2$ is called upon, the proposed method will search through a series of adjacent points, starting from $\boldsymbol{C}_{n-1}$ in the $+\boldsymbol{D}_2$ forward direction to determine the NTP for $\boldsymbol{P}_n$. As a result, it locates the NTP $\boldsymbol{C}_n$ still the previous $\boldsymbol{C}_{n-1}$ for $\boldsymbol{P}_n$, leading to smooth force rendering. Based on the discussions above, we develop Algorithm 1, which yields the searching process for locating the NTP $\boldsymbol{C}_n$ for $\boldsymbol{P}_n$ by the previous NTP $\boldsymbol{C}_{n-1}$ and the line tool $+\boldsymbol{D}$. Algorithm 1 can be extended for the bidirectional line tool by taking $+\boldsymbol{D}$ for forward movement and $-\boldsymbol{D}$ for backward. As for the plane tool, it will take two of those for the line tool, as its search of NTP will be in two DOF. And, for the point tool, the desired point itself is the NTP, which can be used for computation in Equation (3) directly. To satisfy the demand for haptic rendering, δ is set to be 0.1 mm and haptic updating frequency about 1 kHz.

Algorithm 1: Locating NTP $\boldsymbol{C}_n$ by $\boldsymbol{P}_n$, $\boldsymbol{C}_{n-1}$, and $+\boldsymbol{D}$

```
let C' = C_{n-1} + δ(+D)
if ||P_n − C'|| < ||P_n − C_{n-1}|| then
    C_{n-1} ⇐ C'
    go to 1
end if
C_n ⇐ C_{n-1}
```

3. System implementation

Figure 7 shows the system implementation block diagram of the proposed manipulation system. We adopt the 6-DOF Mitsubishi RV-2A robot manipulator for manipulation, with its joint specification listed in Table 1. In Figure 7, the user sends in the commanded force $\boldsymbol{f}_h$ and receives the reflected force $\boldsymbol{f}_r$ from the 6-DOF haptic device (Phantom Omni with 3-DOF force feedback and maximum force of 3.3 N, manufactured by SensAble technology, Wilmington, MA, USA). $\boldsymbol{f}_h$ and $\boldsymbol{f}_r$, combined into $\boldsymbol{f}_u$, are sent to the haptic device, which generates the command $\boldsymbol{x}$. This command $\boldsymbol{x}$, transformed into joint command $\boldsymbol{q}$ by the command processor, is then forwarded to the robot manipulator via the local area network. Meanwhile, the robot status $\boldsymbol{x}_r$ is fed back to the manipulation system, becoming $\boldsymbol{v}_s$ via the visual engine and $\boldsymbol{f}$ via the haptic engine, to generate both visual and haptic effects. This $\boldsymbol{f}$ further becomes $\boldsymbol{f}_r$ via the haptic device. The stiffness values for the virtual spring and virtual tools are empirically determined to be 0.08 and 0.8 N/mm, respectively. To satisfy the requirements from various devices, we use three threads (under Intel Core Duo E8400 CPU and 2 GB RAM) to deal with the

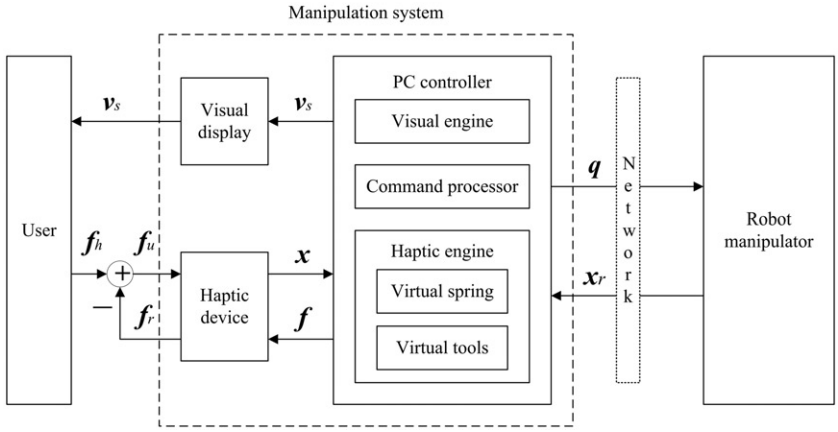

Figure 7. System implementation block diagram of the proposed manipulation system.

Table 1. Joint specifications of Mitsubishi RV-2A robot arm.

Joint	Operating range (°)	Speed of motion (°/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

haptic data (about 1 kHz), visual data (about 60 Hz), and manipulator control data (about 141 Hz). For matching up the workspaces between the haptic device and robot manipulator, the scaling factor k in Equation (1a) is set to be 2. And, the 3-D graphical environment is constructed using OpenGL, rendering the virtual robot and tool in a real-time manner. The graphical environment also yields color change and sound warning for cases when the robot joint exceeds the operating range or the obstacle, e.g., the ground, is close by. When the command processor detects such kinds of events, it halts the robot for safety.

4. Experiments

For performance evaluation, we apply the proposed manipulation system to conduct a liquid-pouring task in a chemical laboratory environment, during which the robot manipulator is governed to grasp and move a tube (tube A) to pour the liquid inside into another tube (tube B). Figure 8 shows the experimental setup, in which the two tubes, with their top part of the diameter and length at 15.7 and 120 mm, respectively, are placed in the holes of a rack. These two holes, separated by a distance of 200 mm, are with the diameter and depth at 15.7 and 75 mm, respectively. Due to the closeness of the diameters of the tube and hole and the nature of liquid pouring, this task demands very delicate maneuvering from the experimenter. The experimental procedure is divided into the seven phases below:

Phase 1: locate tube A (with the liquid), move the robot manipulator, close its gripper to pick up tube A from the hole, and keep it at a proper height.
Phase 2: locate tube B (for liquid pouring) and move tube A to be on top of tube B along a horizontal plane.
Phase 3: rotate tube A to make its opening aligned with that of tube B.
Phase 4: rotate tube A further around a fixed point to empty all the liquid into tube B.
Phase 5: rotate tube A back and keep it at a proper height.
Phase 6: move tube A back to be on top of its original hole along a horizontal plane.
Phase 7: place tube A into the hole and open the gripper of the robot manipulator.

During task execution, the experimenter can determine when and where to execute each of the phases sequentially. A successful trial is defined to be at least 10 mL out of the initial 11 mL liquid in tube A has been poured into tube B; otherwise, it is taken as a failure. Corresponding to these seven phases, the virtual tools can be called for assistance in manipulation, described below:

Phase 1: the line tool for assistance in a straight-up motion;
Phase 2: the plane tool for movement in a horizontal plane;
Phase 4: the point tool for a fixed point rotation;

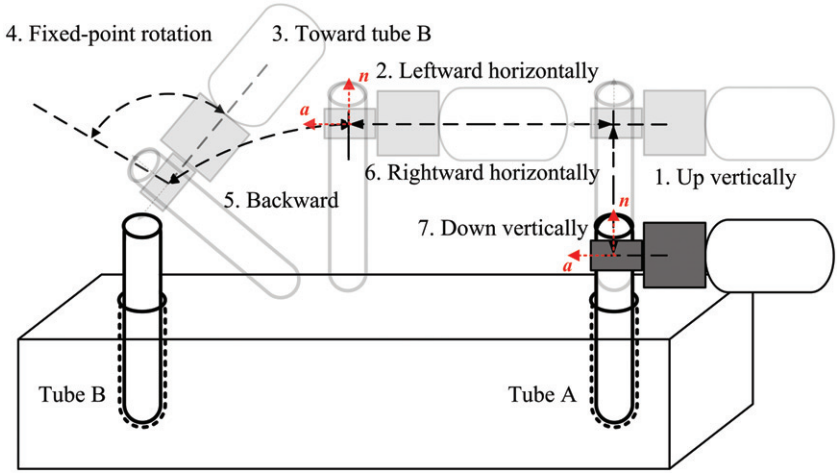

Figure 8. Experimental setup.

Phase 5: the plane tool for movement in a vertical plane;
Phase 6: the plane tool for movement in a horizontal plane; and
Phase 7: the line tool for assistance in a straight-down motion.

Note that there is no virtual tool provided for phase 3, because it demands freedom for the user to align the opening of tube A to that of tube B. In addition to the virtual tools, we also provide the option of including the effect of the virtual spring (discussed in Section 2.1) for the experiments. Thus, totally four operation modes are provided for the experimenter: NS-NA (no virtual spring, no virtual tool), S-NA (with virtual spring, no virtual tool), NS-A (no virtual spring, with virtual tool), and S-A (with virtual spring, with virtual tool). Six male and two female graduate students from our institute, between 23 and 24 years old, all right-handed, were invited to conduct the experiments. Only two of them had experience in manipulating the robot manipulator using the haptic device before, while the other six had little or no experience. To alleviate the learning effect, the order of these four operating modes was randomly arranged for each subject, and they were asked to achieve at least two successful trials for each mode. Before conducting the trial for each mode, the subject practiced several times to be familiar with the manipulation system.

Table 2 shows the results related to the number of failures corresponding to the four operation modes. In Table 2, mode NS-NA led to the highest number of failures, and modes S-NA, NS-A, and S-A all led to low numbers of failures, suggesting that the virtual spring and virtual tools did provide the assistance. Figure 9 shows the time box plots for the execution time for the entire process and also phases 4 and 7, which pose more challenges to the subjects. In Figure 9(a), according to the medians for the four modes (NS-NA: 59.2 s, S-NA: 59.9 s, NS-A: 53.5 s, and S-A: 50.0 s), the virtual tools reduced the execution time up to about 9.9 s for mode S-A versus S-NA and 5.7 s for mode NS-A versus NS-NA. The reduction in execution time using the virtual tools is also evident in Figure 9(b) and (c) for phases 4 and 7. Figure 10 shows the traveling path box plots for the robot manipulator for the entire process and also for phases 4 and 7. All Figure 10(a)–(c) show that mode S-A led to the smallest median, indicating that the use of the virtual tools and virtual spring did shorten the traveling path. For phase 4 in Figure 10(b), the point tool provided much help for the subject to rotate tube A around a fixed point for pouring. To analyze the individual effects of the virtual spring and virtual tools on the performance, according to the medians in Figure 10(a) for the four modes (NS-NA: 817 mm, S-NA: 778 mm,

Table 2. Number of failures for the four operation modes.

	Operation mode			
User	NS-NA	S-NA	NS-A	S-A
A	0	0	0	0
B	1	1	0	0
C	1	0	0	0
D	2	0	1	1
E	0	0	1	0
F	0	1	0	1
G	2	0	1	0
H	0	0	0	0
Sum	6 (27%)	2 (11%)	3 (16%)	2 (11%)

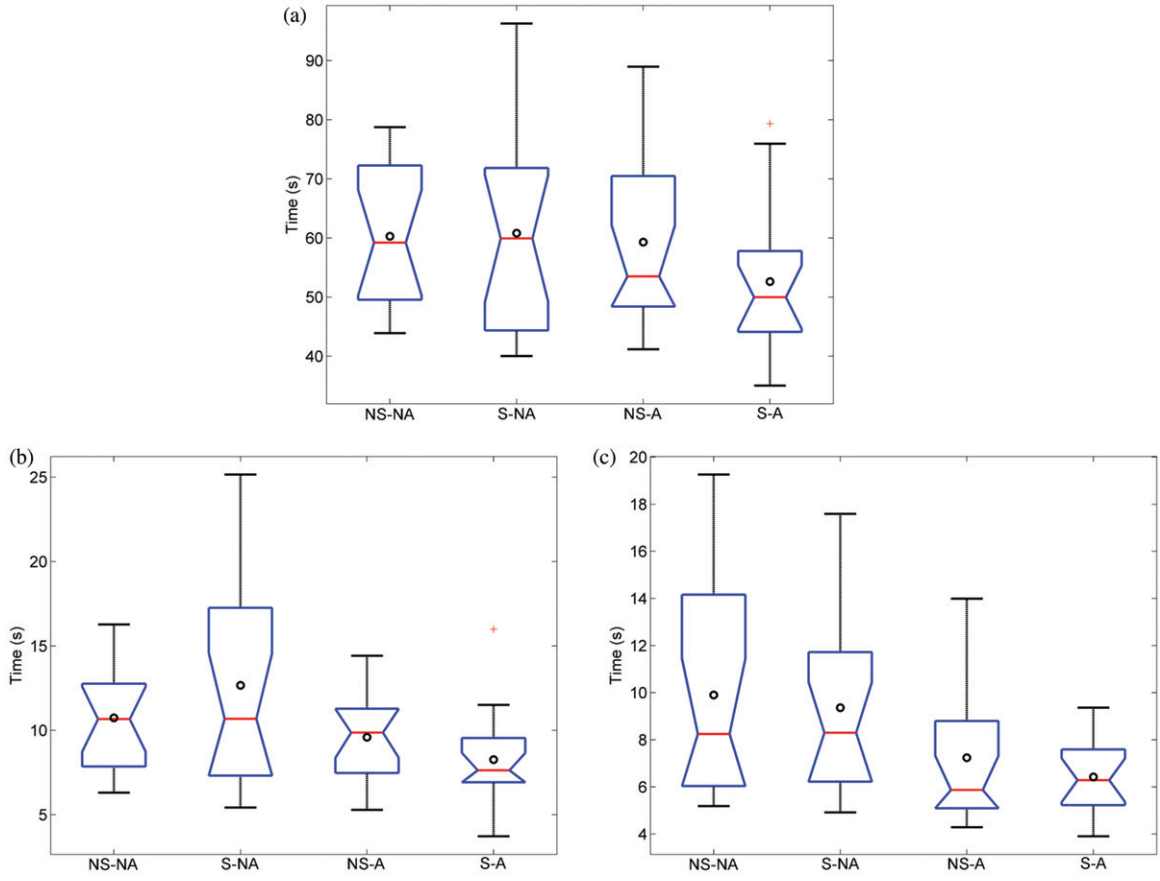

Figure 9. Time box plots for: (a) the entire process, (b) phase 4, and (c) phase 7.

NS-A: 760 mm, and S-A: 721 mm), the virtual spring reduced the traveling path up to about 39 mm for mode S-NA versus NS-NA and also mode S-A versus NS-A, and the virtual tools up to about 57 mm for mode NS-A versus NS-NA and also mode S-A versus S-NA. We further look into the entire trajectories for the four modes during task execution. Figure 11 shows those trajectories for user H as an example. In Figure 11(c) and (d) for trajectories involving the use of the virtual tools (modes NS-A and S-A), the virtual tools led to quite straight movements in phases 1 and 7, more smooth movements in a horizontal plane in phases 2 and 6, and steady rotation in phase 4. After the completion of the experiments, the subjects were asked to vote their preference on the operation modes, as listed in Table 3. In Table 3, mode S-A is voted to be the most preferred operation mode and mode NS-NA the least. Most subjects reported that the bonding forces from the virtual spring and virtual tools did provide support to maintain stable movements. And, the most difficult operation for several subjects was to aim at the small target (e.g., the opening of tube B or the hole for tube A on the rack), due to quite a distance present between them and the manipulated object.

5. Conclusion

In this article, we have proposed an effective manipulation system for multi-DOF robot manipulators based on virtual linkage and virtual tools. The virtual linkage enhances the connection between the user and robot manipulator, and virtual tools are helpful in dealing with the requirements of various tasks. We have applied the proposed manipulation system to conduct experiments in the chemical laboratory environment, which demand delicate maneuvers. Satisfactory experimental results and analyses of users' responses have verified its effectiveness. In future works, we plan to improve the proposed manipulation system by including torque feedback, in addition to current force feedback, so that the applications can be extended for tasks that demand assistance in orientation governing. Another possible improvement lies in the introduction of the damping

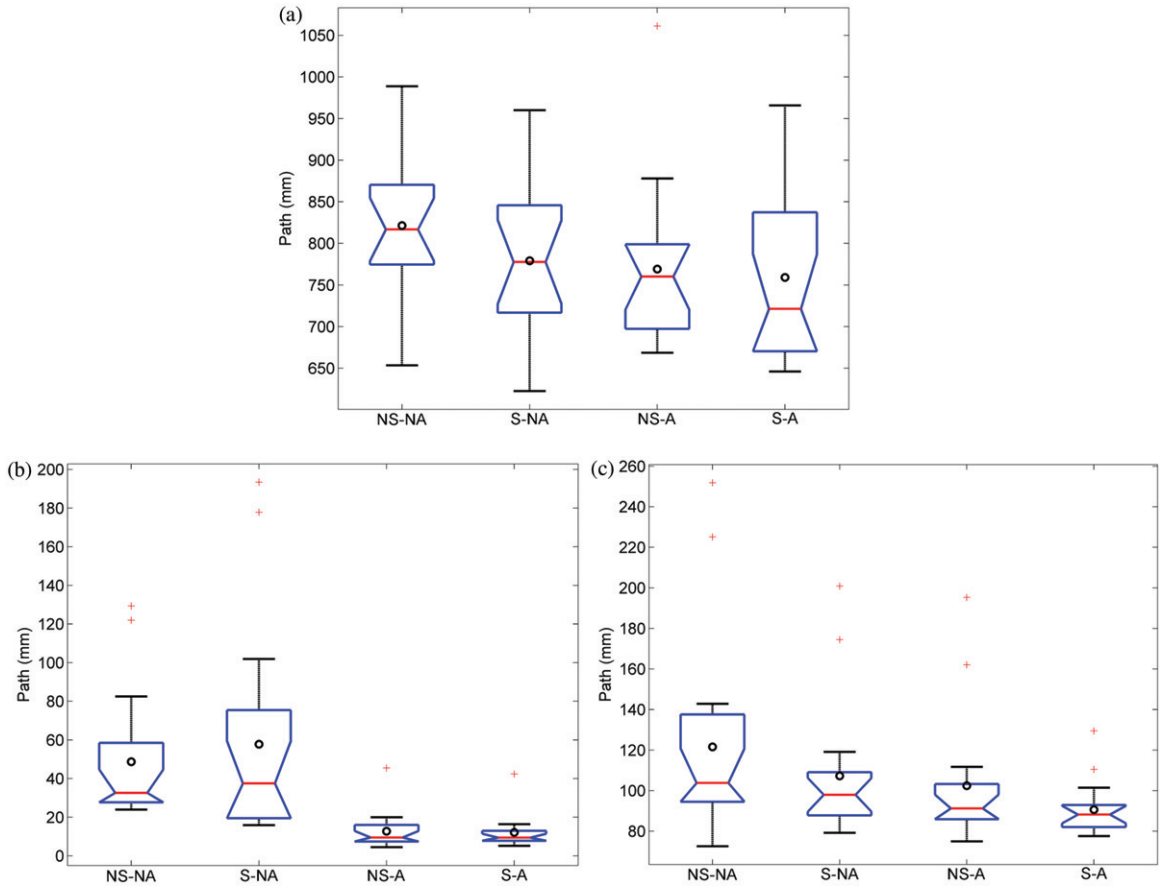

Figure 10. Traveling path box plots for: (a) the entire process, (b) phase 4, and (c) phase 7.

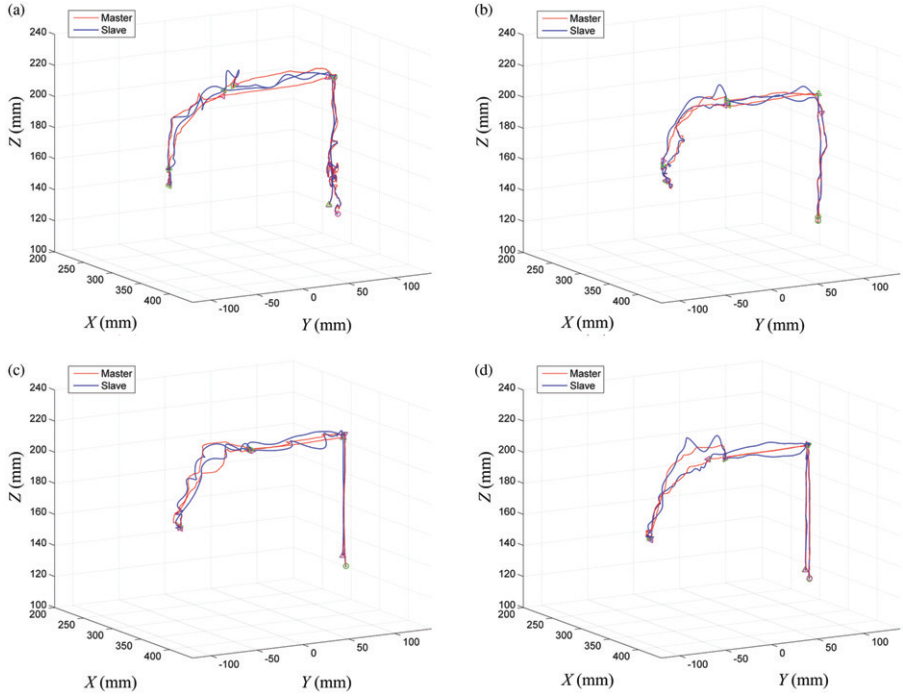

Figure 11. Trajectories of user H in modes: (a) NS-NA, (b) S-NA, (c) NS-A, and (d) S-A.

Table 3. Operation mode preference.

	Operation mode			
Preference order	NS-NA	S-NA	NS-A	S-A
First	0	1	0	7
Second	0	1	6	1
Third	1	6	1	0
Fourth	7	0	1	0

effect, along with the spring effect, for better human–robot interaction.

Acknowledgments

Part of this article has been presented at International Symposium on Artificial Life and Robotics, January 19–21, 2012, Beppu, Japan. This study was supported in part by the National Science Council under grant NSC 99-2221-E-009-157, and also by the Department of Industrial Technology under grant 97-EC-17-A-02-S1-032.

Nomenclature

$\boldsymbol{C}_M$, $\boldsymbol{C}_S$ the joystick workspace center and the robot manipulator workspace center, see Equation (1)

$\boldsymbol{C}_n$, $\boldsymbol{P}_n$ the NTP on the virtual tool and its corresponding tip position of the haptic device, see Equation (3)

$\boldsymbol{F}_n$ the constrained force of the virtual tool, see Equation (3)

$\boldsymbol{F}_s$ the constrained force of the virtual spring, see Equation (2)

K the stiffness of the virtual tool, see Equation (3)

k the scaling factor for matching up the size of workspaces of the joystick and robot manipulator, see Equation (1)

K_s the stiffness of the virtual spring, see Equation (2)

$\boldsymbol{P}_M$, $\boldsymbol{R}_M$ the position and orientation in the coordinate system of the joystick, see Equation (1)

$\boldsymbol{P}_r$, $\boldsymbol{P}_h$ the end-effector position and the force-reflection joystick position, see Equation (2)

$\boldsymbol{P}_S$, $\boldsymbol{R}_S$ the position and orientation in the coordinate system of the robot manipulator, see Equation (1)

$\boldsymbol{R}_m^s$ the rotational matrix from the coordinate system of the joystick to that of the robot manipulator, see Equation (1)

References

Abbott, J.J., Marayong, P., and Okamura, A.M., 2007. Haptic virtual fixtures for robot-assisted manipulation. *In*: S. Thrun, R. Brooks and H. Durrant-Whyte, eds. *Robotics research*. Berlin: Springer, 49–64.

Forsyth, B.A. and MacLean, K.E., 2006. Predictive haptic guidance: intelligent user assistance for the control of dynamic tasks. *IEEE transactions on visualization and computer graphics*, 12 (1), 103–113.

Glover, C., *et al*., 2009. An effective and intuitive control interface for remote robot teleoperation with complete haptic feedback. *In*: *Emerging technologies conference*, 2–3 April 2009, Ames, IA, Iowa State University.

Guo, C. and Sharlin, E., 2008. Exploring the use of tangible user interfaces for human-robot interaction: a comparative study. *In*: *SIGCHI conference on human factors in computing systems*, 5–10 April 2008, Florence, Italy. New York: ACM, 121–130.

He, X. and Chen, Y., 2008. Six-degree-of-freedom haptic rendering in virtual teleoperation. *IEEE transactions on instrumentation and measurement*, 57 (9), 1866–1875.

He, X. and Chen, Y., 2009. Haptic-aided robot path planning based on virtual tele-operation. *Robotics and computer-integrated manufacturing*, 25 (4–5), 792–803.

Hsieh, M.-C. and Young, K.-Y., 2010. Motion constraint design and implementation for a multi-functional virtual manipulation system. *Mechatronics*, 20 (3), 346–354.

Joly, L. and Andriot, C., 1995. Imposing motion constraints to a force reflecting telerobot through real-time simulation of virtual mechanisms. *In*: *IEEE international conference on robotics and automation*, 21–27 May 1995, Nagoya, Aichi, Japan. Piscataway, NJ: IEEE, 357–362.

Kemp, C.C., Edsinger, A., and Torres-Jara, E., 2007. Challenges for robot manipulation in human environments. *IEEE robotics and automation magazine*, 14 (1), 20–29.

Kuan, C.-P. and Young, K.-Y., 2001. VR-based teleoperation for robot compliance control. *Journal of intelligent and robotic systems*, 30 (4), 377–398.

Kuang, A.B., *et al*., 2004. Assembling virtual fixtures for guidance in training environments. *In: International symposium on haptic interfaces for virtual environment and teleoperator systems*. 27–28 March 2004, Chicago, IL. Piscataway, NJ: IEEE, 367–374.

Lai, L.-C., Wu, C.-J., and Shiue, Y.-L., 2007. A potential field method for robot motion planning in unknown environments. *Journal of the Chinese institute of engineers*, 30 (3), 369–377.

Li, M. and Okamura, A.M., 2003. Recognition of operator motions for real-time assistance using virtual fixtures. *In*: *International symposium on haptic interfaces for virtual*

environment and teleoperator systems, 22–23 March 2003, Los Angeles, CA. Piscataway, NJ: IEEE, 125–131.

Marayong, P., *et al*., 2003. Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures. *In*: *IEEE international conference on robotics and automation*, 14–19 September 2003, Taipei, Taiwan. Piscataway, NJ: IEEE, 1954–1959.

Nahvi, A., *et al*., 1998. Haptic manipulation of virtual mechanisms from mechanical CAD designs. *In*: *IEEE international conference on robotics and automation*, 16–20 May 1998, Leuven, Belgium. Piscataway, NJ: IEEE, 375–380.

Park, H.S., *et al*., 2005. HMM-based gesture recognition for robot control. *In*: J. Marques, N. Pérez de la Blanca and P. Pina, eds. *Pattern recognition and image analysis*. Berlin: Springer, 607–614.

Peshkin, M.A., *et al*., 2001. Cobot architecture. *IEEE transactions on robotics and automation*, 17 (4), 377–390.

Prada, R. and Payandeh, S., 2005. 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*, 18–20 March 2005, Pisa, Italy. Piscataway, NJ: IEEE, 375–380.

Prada, R. and Payandeh, S., 2009. On study of design and implementation of virtual fixtures. *Virtual reality*, 13 (2), 117–129.

Preusche, C., Ortmaier, T., and Hirzinger, G., 2002. Teleoperation concepts in minimal invasive surgery. *Control engineering practice*, 10 (11), 1245–1250.

Rosenberg, L.B., 1993. Virtual fixtures: perceptual tools for telerobotic manipulation. *In*: *IEEE virtual reality annual international symposium*, 18–22 September 1993, Seattle, WA. Piscataway, NJ: IEEE, 76–82.

Ruspini, D.C., Kolarov, K., and Khatib, O., 1997. The haptic display of complex graphical environments. *In*: *ACM international conference on computer graphics and interactive techniques*, 3–8 August 1997, Los Angeles, CA. New York: ACM, 345–352.

Song, G., Guo, S., and Wang, Q., 2006. A tele-operation system based on haptic feedback. *In*: *IEEE international conference on information acquisition*, 20–23 August 2006, Weihai, Shandong, China. Piscataway, NJ: IEEE, 1127–1131.

Stanczyk, B., Peer, A., and Buss, M., 2006. Development of a high-performance haptic telemanipulation system with dissimilar kinematics. *Advanced robotics*, 20 (11), 1303–1320.

Veras, E., *et al*., 2009. Laser-assisted real-time and scaled telerobotic control of a manipulator for defense and security applications. *In*: *SPIE defense, security and sensing conference*, 14–17 April 2009, Orlando, FL. Bellingham, WA: SPIE, 73321S.

Yu, W., Dubey, R., and Pernalete, N., 2004. Telemanipulation enhancement through user's motion intention recognition and fixture assistance. *In*: *IEEE/RSJ international conference on intelligent robots and systems*, 18 September – 2 October 2004, Sendai, Japan. Piscataway, NJ: IEEE, 2235–2240.

Zilles, C.B. and Salisbury, J.K., 1995. A constraint-based god-object method for haptic display. *In*: *IEEE international conference on intelligent robots and systems*, 5–9 August 1995, Pittsburgh, PA. Piscataway, NJ: IEEE, 146–151.