Control of a Telemanipulator using Freehand and Force/Visual Feedback

Nopasorn Kowathanakul, Jaruboot Kananai, and Ratchatin Chancharoen*

Mechanical Engineering Department, Chulalongkorn University, Bangkok, 10330, Thailand
*Corresponding Author: Ratchatin.C@chula.ac.th, 662-2186643, 662-2522889

Abstract

In this project, the motion of a telemanipulator is controlled by using only freehand. The telemanipulator system uses Kinect sensor to approximately sense the 3D position of an operator's hand which is used to generate the command signal to control the motion of a telemanipulator. The telemanipulator is equipped with a camera and force sensor in its hand and the RGB video and force signal are sent back to the operator in real time. In this problem, we have to face several difficulties including various delay and lost package in communication between operator and telemanipulator, uncontrolled and various sampling of hand position, and high uncertainty in the hand position signal. A robust and practical controller is design to cope with all these difficulties. The control algorithm is derived based on feedback linearization technique with dynamics suppression. The experimental results demonstrate that the proposed technique is capable of controlling the telemanipulator such that we use only freehand to smoothly control its motion.

1. Introduction

A telemanipulator, or somehow called remoted manipulator, is a device for transmitting hand and/or finger movements to a remote robotic device, allowing the manipulation of objects that are too heavy, dangerous, small, or otherwise difficult to handle directly [1]. Consider a typical bilateral haptic telemanipulator as shown in Fig. 1 [2], in which an operator controls the motion of a telemanipulator via a haptic interface. This haptic interface senses the motion of the operator's hand or finger then informs the master control. Consequently, the master control generates a command signal and sends it to the telemanipulator control via communication channel. The telemanipulator control regulates the motion of the telemanipulator in accordance with this command and also sends the perception from the telemanipulator back to the operator. The control techniques for a telemanipulator are widely researched [3, 4, 5].

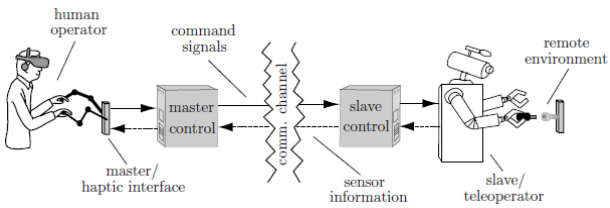


Fig. 1 Typical Bilateral Haptic Telemanipulator [2]

The performance of a telemanipulator system includes robustness, task performance, feeling of presence, and transparency as mentioned in [2]. The system contains uncertainties from various sources but must be vigorously stable. Furthermore, the telemanipulator must be able to comply with a desired task in a remote environment. The operator should be able to feel a remote environment as he/she is being there during operation. Transparency is also desired which mean that the dynamic of technical medium is to be suppress as much as possible.

In addition, the joystick or haptic device such as Sensable's Phantom Omni [6] can also be used as a master device. They contain mechanical links which is able to transmit the motion of an operator to the position sensors and transmit forces back to the operator. In this project, the non-contact Kinect sensor is used as a master device to provide the most natural interaction with a telemanipulator. In the proposed system, the motion of an operator's hand is sensed by using Kinect camera and the live video, from the camera installed on the telemanipulator hand and sent back directly to the operator. Only a time delay is thus felt since there is no mechanical link touching the operator. The robust and practical trajectory controller is also proposed to control the telemanipulator to perfectly follow the trajectory command from the master with a fixed step delay. It is noted that this delay will be reduced when the speeds of the signal processor and communication increases.

2. The Master Slave System

2.1 Kinect and image processing technique

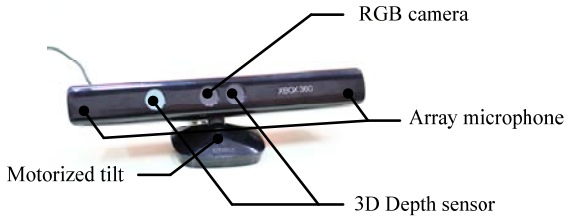
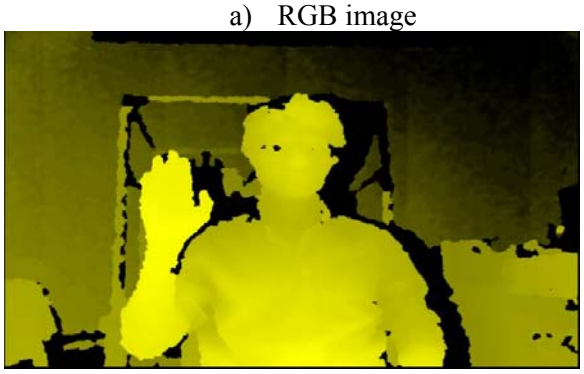


Fig. 2 Kinect

Kinect, launched in 2010, is an interface device for the Microsoft's Xbox 360 video game console. It consists of the RGB camera, depth sensor, multi-array microphone, and motorized tilt (Fig. 2). The camera and the depth sensor is able to output the 8-bit RGB and 11-bit depth images with 640x480 resolutions at the speed up to 30 fps respectively. Both images can be combined and treated as a single image with four dimensions called RGBD (Red, Blue, Green, and Depth) image. The depth image significantly simplifies the segmentation tasks, especially when an object's color is close to an environment. Using Kinect, the object and background with the same color can be easily distinguished by their depths and the RGB image can be processed further to extract the useful information of an object such as the centroid of a hand.

a) RGB image



b) Depth image

Fig. 3 RGB and Depth images from Kinect

By processing an RGBD image, we can determine the 3D motion of the user. In this project, the 3D position of the user's hand (centroid) is captured and used to control the telemanipulator. Kinect also has a microphone array for voice command. This microphone array outputs 16-bit audio at 16 kHz. However, this is not used in this stage of the project. As a effective and affordable device, Kinect is fascinated for robotics education [7].

The C++ program uses the modules from Prime Sensor™ NITE 1.3 Algorithms [8] to acquire images, track operator's hand, and determine its 3D position in real time. The program sends the 3D position over the Ethernet to a telemanipulator via TCP/UDP using an Open Sound Control (OSC) library. In this way, we can use free hand to generate the 3D position reference command in real time and send it to operate the motion of a telemanipulator.

As the sample of raw RGB and depth images are shown in Fig. 3, the object can be identified by its color and depth. Segmentation and skin detection is significantly simplified with depth information, thus an operator's hand is precisely tracked as can be seen as Fig. 4.

Fig. 4 Hand tracking

2.2 CRS Robotics manipulator

The articulated CRS A255 robot, shown in Fig. 5, is modified such that the joint level control algorithm can be implemented. The DH parameter for this robot is given in Table 1. The commercial controller is replaced with our designed controller that can control joint torque for all axes simultaneously at 1 kHz sampling rate. The Matlab® xPC real-time target PC is used as real-time controller. The program is developed in Matlab® Simulink on the host PC and directly uploaded to the target PC for real-time execution via TCP/IP. The target PC is driven by Pentium processor with 1 GB RAM and installed with PCI-8133 encoder card (Adlink Technology Inc.) and PCL-726 D/A card (Advantech Inc.). The

Copley Controls motor amplifiers are used to amplify the power and control the current to each motor. The drivers for all cards are developed in the project for the best performance.

Fig. 5 the CRS Robotics Manipulator.

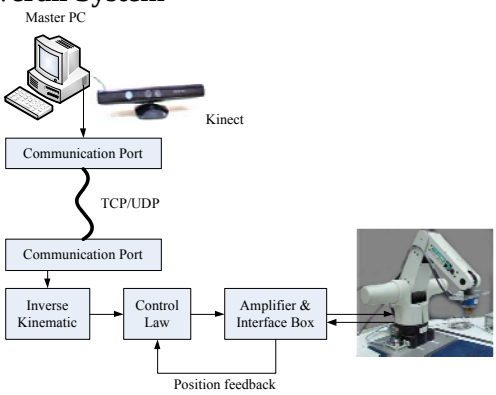
Table 1 DH parameter of the manipulator

i	α_i	a_i	d_i	θ_i
1	0	0	10”	θ_1
2	-90°	0	0	θ_2
3	0	10”	0	θ_3
4	0	10”	0	θ_4
5	-90°	0	2”	θ_5

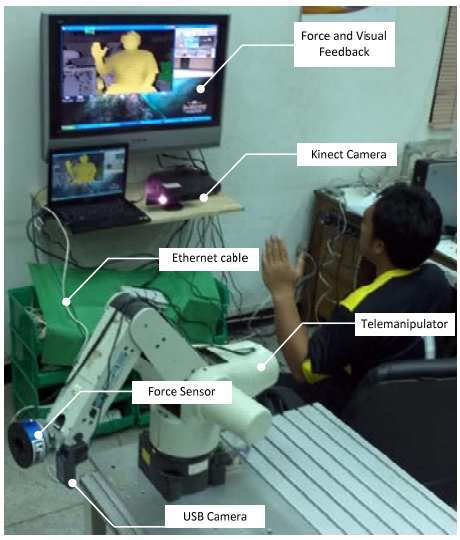
2.3 Hardware Processors

In order to receive a robust signal from Kinect camera, a C++ program, running on Microsoft Windows XP with the Kinect driver installed and Prime Sensor™ NITE library, is written to capture the signals from the Kinect camera and do hand tracking. Then, the position of an operator’s hand is sent over TCP/UDP network to the telemanipulator. The Intel® Core™2 Duo processor notebook with 2GB RAM is used for the master while the Intel® Pentium PC is used to receive the position command from TCP/UDP and to control the motion of the telemanipulator. The Matlab Xpc target is used as a real-time controller for the telemanipulator.

2.4 Overall System



a) The signal flow diagram



b) The real system

Fig. 6 the Master Slave System

An operator uses his left hand to control the motion of the CRS manipulator without directly seeing the robot. The feedback video (small upper left window on the TV screen) from the USB camera at the robot’s hand is guide for the operator incase that he move robot in free space.

3. Control Technique

3.1 Control design

The inverse ARX controller is developed in the project to effectively control the motion of a telemanipulator. The controller is designed to cope with several delay and lost package in communication since the communication channel is standard TCP/IP network. We proposed that the telemanipulator’s controller should be able to suppress not only the nonlinearity but also the dynamic behavior. Consider the dynamical equation of motion [9] of the manipulator

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta,\dot{\Theta})\dot{\Theta} + G(\Theta) \qquad (1)$$

The *M*, *V*, and *G* are assumed to be constant and not *a priori* known. Consider one joint of the manipulator that is to be controlled using digital controller. The sampled data control system is modeled as in Fig. 7.

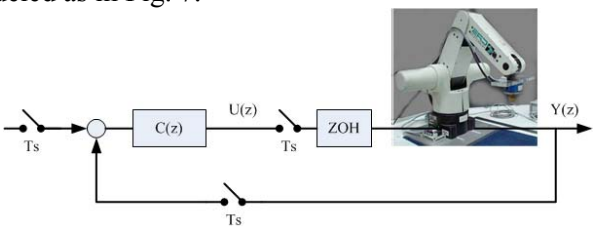


Fig. 7 The sampled telemanipulator system

The model of one joint can be written in Laplace domain and *z*-domain (zero order hold) as

$$\frac{\theta(s)}{T(s)} = \frac{1}{Ms^2 + Vs + G} \qquad (2)$$
$$\frac{\theta(z)}{T(z)} = \frac{B(z)}{A(z)} = c\frac{z + b_0}{z^2 + a_1 z + a_0}$$

The *z*-domain model can be written in ARX (Autoregressive with eXogeneous input) model [10]. The ARX model is a polynomial model uses a generalized notion of transfer functions to express the relationship between the input, *u(t)*, the output *y(t)*, and the noise *e(t)*. The benefit is that the variables in the model are polynomials expressed in the time-shift operator q^{-1}. The variables can be fitted using the generated data pairs. The general form of an ARX model is

$$A(q)y(k) = B(q)u(k - n_k) + e(k) \qquad (3)$$

where $A(q) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n_a}$

$B(q) = b_1 + b_2 q^{-1} + \ldots + b_{n_b} q^{-n_b+1}$

q^{-1} is the time-shift operator.

n_k is the time delay.

e is the error.

y and u are θ and τ respectively

Thus, one joint can be modeled as ARX using n_a, n_b, and n_k = 3, 2, and 1 respectively. The ARX model for one joint is then written as

$$y(k) + a_1 y(k-1) + a_2 y(k-2) = b_1 u(k-1) + b_2 u(k-2) \qquad (4)$$

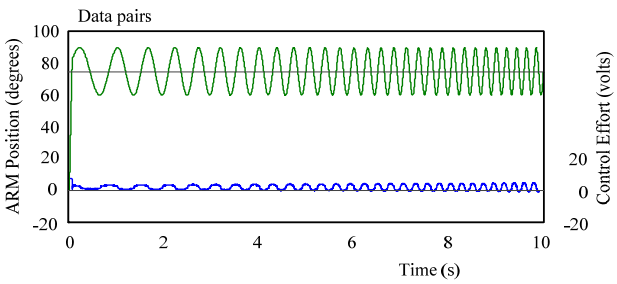


Fig. 8 the data pairs to fit the ARX model

Consider the elbow joint of this robot. We generated the data pairs by controlling the elbow joint using a stiff PD to track the sine sweep signal -15±45 degrees from 0.1-0.5 Hz in 100 seconds. The data pairs are as shown in Fig. 8. These data pairs are used to fit all the coefficients in the ARX model.

3.2 Control Algorithm

The model in (4) is shifted by one time step to become

$$y(k+1) + a_1 y(k) + a_2 y(k-1) = b_1 u(k) + b_2 u(k-1) \qquad (5)$$

Since we know *y* and *u* at time step *k* and their past values but not the future value. The model in (5) is rearranged such that we can find *u(k)* to give *y(k+1)* as *y*(k)*. Therefore, the control law becomes

$$u(k) = \frac{y^*(k) + a_1 y(k) + a_2 y(k-1) - b_2 u(k-1)}{b_1} \qquad (6)$$

where y^* is the reference position. Then, substitute *u(k)* in (6) into (5) results in

$$y(k+1) = y^*(k) \qquad (7)$$

As a result, the output will beat the reference with one step delay. This is deadbeat controller that is designed to beat a reference within a certain time step. However, its control effort is extremely high and chattering. To smoothen the chattering, the control must be less exacting. The filter is designed to convert a deadbeat controller into Dahlin controller.

$$F(z) = \left[\frac{1 - z^{-k}}{z^{-k}}\right]\left[\frac{(1-q)z^{-1}}{1 - qz^{-1} - (1-q)z^{-k}}\right] \qquad (8)$$

where q = $exp(-Ts/\tau)$

Ts is the sample time.

τ is the time constant.

k is a deadbeat time step

With the filter, the control will not beat the output in one step but soften follow the reference with τ time constant. The close loop system behavior like a first order plus time delay system and the control effort is significantly reduced and chattering is suppressed.

4. Problem Statement

An operator controls the motion of the telemanipulator by using one hand while he does not directly seeing the manipulator. The visual and force feedback from the manipulator is provided to guide the operator if he tries to move the manipulator in free space. The positions of the telemanipulator and the reference command are recorded to evaluate the performance of the system. The feeling of an operator when using the system is also recorded.

5. Experimental Result

5.1 Task Performance Evaluation

Several studies on telemanipulation attempted to evaluate the performance of the system based on task. Performance metrics normally include task completion time, operator subjective assessment, position error, joint effort and etc [12]. In this stage of the project, we evaluate how well the telemanipulator can track the motion of an operator's hand. The error between the reference and actual trajectories is reported in this paper. Fitt's Law [13] is under studied for the next stage to evaluate the task performance

5.2 Results

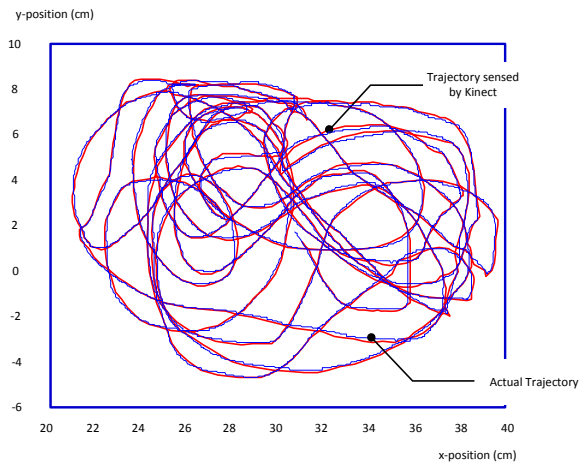


Fig. 9 Main Results

In the experiment, an operator controlled the motion of the robot by his left hand. He randomly moved his hand in a vertical plane. The position, (x,y,z), of the hand is sensed by Kinect camera and sent to the telemanipulator via TCP/UDP. The sampling time is not guaranteed. The telemanipulator is controlled to track this command position in real time. The force sensors and USB camera sent the force signals and real-time video back to the operator. The video on the right of the screen shows live video back from the telemanipulator. The trajectory of the telemanipulator is quite smooth and close to the trajectory of the operator's hand. However, an operator still experiences a time delay which we are now working on to cope with this.

6. Future Work

The system is now re-architectured to effectively handle all the signals at the higher speed. The master computer is now upgraded to Windows 7 (from Windows XP) and thus all the libraries and drivers have to be reinstalled and tested. The program, to convert the force feedback into to visual display, which is then overlaid on the live video, is under developed. Furthermore, task performance evaluation based on Fitt's Law is under studied and will be used to evaluate the task performance in the next stage of the project.

7. Acknowledgement

This research is sponsored by the Chulalongkorn university century academic development project.

8. References

[1] Guittet, J and Parent, M. (1979). Programming automatic reflex actions in telemanipulation. *Decision and Control including the Symposium on Adaptive Processes, 1979 18th IEEE Conference*, vol. 18, December 1979, pp. 1021-1023.

[2] Passenberg, C., Peer, A. and Buss, M. A survey of environment-, operator-, and task-adapted controllers for teleoperation systems, *Mechatronics*, Volume 20, Issue 7, October 2010, pp. 787–801.

[3] Franken, M., Stramigioli, St., Misra, S., Secchi, C. and Macchelli, A. Bilateral Telemanipulation With Time Delays: A Two-Layer Approach Combining Passivity and Transparenc, *IEEE trans on robotics*, vol. 27, no. 4, August 2011.

[4] Tavakoli, M., Aziminejad, A., Patel, R. V., and Moallem, M. High-Fidelity Bilateral Teleoperation Systems and the Effect of Multimodal Haptics, *IEEE trans on SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, vol. 37, no. 6, December 2007.

[5] Kim, J.-P., Seo, C. and Ryu, J. Multirate Energy-Bounding Algorithm for High-Fidelity Stable Haptic Interaction Control, *IEEE trans on CONTROL SYSTEMS TECHNOLOGY*, vol. 19, no.5, September 2011.

[6] Jarillo-Silva, A., Domínguez-Ramírez, O.A., Perra-Vega, V. and Ordaz-Oliver, J.P. (2009). PHANTOM OMNI Haptic Device: Kinematic and Manipulability, *Electronics, Robotrics, and Automotive Mechanics Conference 2009*, September 2009, pp. 193-198.

[7] Tölguessy M. and Hubinsky, P. (2011). The Kinect Sensor in Robotics Education in the *Proceeding of 2nd International Conference on Robotics in Education (RiE2011)*, Vienna, Austria.

[8] PrimeSense Limited Company. *Prime Sensor™ NITE 1.3 Algorithms notes*, URL: http://www.primesense.com

[9] John J. Craig (1989). Introduction to ROBOTICS: mechanics and control, Addison-Wesley Publishing Company, ISBN: 0-201-09528-9, USA.

[10] Oliver Nelles (2001). Nonlinear system identification: from classical approaches to neural networks and fuzzy models, ISBN: 3-540-67369-5, Springer-Verlag, Berlin, Heidelberg, New York.

[11] Zhi-Gabg, Z., Ben-Guo, Z. and Zhen-Fu, B. (2009). Dahlin algorithm design and simulation for time-delay system, *Control and Decision Conference 2009*, June 2009, pp. 5819-5822.

[12] McLean, G.F., Prescott, B. and Podhorodeski, R (1994). Teleoperated System Performance Evaluation, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 5, May 1994, pp. 796-804.

[13] Mihelj, M., Kamnik, R. and Bajd, T. Assessment of Teleoperator Performance, *Electrotechnical Review*, vol. 65, pp. 7-13, 1998.