Teleoperation of Humanoid Baxter Robot Using Haptic Feedback

Zhangfeng Ju[1,2], Chenguang Yang*[1,3], Zhijun Li[3], Long Cheng[4] and Hongbin Ma*[5]

Abstract— **This paper presents a teleoperation strategy, which is featured by haptic feedback. The teleoperation system is composed of a SensAble® Omni haptic device, set as the master and providing haptic feedback, and an anthropomorphic robot slave, which is embodied by the 7-DOF (degrees of freedom) robotic arm of the Baxter® robot. The haptic feedback enables a bilateral manipulation of the teleoperation system. The joint angles and Cartesian position of the stylus of the Omni device are sampled and transferred to the slave, determining its motion. Meanwhile a force, proportional to the amplitude of position error of the slave manipulator, is sent back to the master and applied to the stylus. Hereby, the operator can sense the motion of the Baxter robot and adjust its manipulator accordingly. The kinematics of the master and slave have been analysed and a workspace mapping has been realized. Two methods, direct angle mapping and CLIK (closed-loop inverse kinematics), are used to implement the manipulation of the slave in position-position mode. Two experiments have been designed and tested to verify the validity of the methods provided by this paper. The results of the experiments illustrate that the designed teleoperation system is feasible and effective.**

I. INTRODUCTION

Robots are used for a large range of different applications in various fields, e.g., industry and medicine. Despite recent advance of technology in the field of autonomous robots, unstructured environments remain a big challenge for machine intelligence. Thus, it is generally believed not feasible for a robot to deal with versatile surroundings autonomously. Therefore, semi-autonomous and human-in-loop methods are preferred for the majority of applications and play important roles for robot manipulators.

The process that an operator control a slave robot remotely is called teleoperation. In recent years, the sector of telemedicine is highly interested in teleoperated robots, since they are able to perform various elaborate tasks including remote surgery and tele-operated rehabilitation. Krebs *et al* suggested a robot which is useful for the neurological rehabilitation therapy for people with immobility of the upper limbs as a result of neurological injuries[1]. Based on virtual reality, Guo *et al* developed a self assisted rehabilitation system for the upper limbs[2]. A lot of different other fields of application have been investigated as well. Li *et al* developed a preliminary teleoperation system based on YARP platform for iCub robot via hand gesture recognition [3]. Inoue *et al* proposed a teleoperated micro hand which consists of two rotational fingers in [4].

Most popular teleoperation system employs the so called master-slave scheme, whereas the master acts as the controller to command the slave and provides an interface to interact with human operator, while the slave executes the commanded task in a significant long distance. The remote information such as position, orientation, force and torque are sampled by the slave and are transferred to the master to enable the human operator react in real time. In most applications, the slave moves to follow the motion of master controlled by a human operator, in a scaled manner. For example, for the teleoperating systems of telesurgery, the slave robot can be controlled to move in the range of minimeter.

Historically, input devices like keyboards and joysticks have been widely used in teleoperating systems. Through the advance of technology, haptic devices moved into the focus of research projects during the last decades. The major advantage of this devices is to provide the operator with a tactile feeling of the remote environment through the force feedback. Thus, the immersion into the virtual reality of the local operation platform can be realized [5]. It has been reported, that the introduction of force feedback into the world of tele-operated systems facilitates the reduction of energy consumption, task completion time and the magnitude of errors [6], [7], [8]. Wang *et al* used a SensAble haptic device to command a mobile robot instead of a robot arm. However, the issues of workspace mapping and inverse kinematics, which are discussed in this paper, were not regarded by their work [9].

In this paper a master-slave teleoperating system using Baxter robot is proposed, which includes haptic feedback. Two preliminary experiments have been performed. The experimental results demonstrate, that the system has a good operationality and could be used in various teleoperational scenarios.

II. SYSTEM DESCRIPTION

The master of the teleoperation system is embodied by the SensAble Omni haptic device, while the slave is realized by Baxter robot of 7-DOF arms. In the 7-DOF robot arm each joint is equipped with a force sensor. The position and force information of the robot's end-manipulator are sampled and sent to the control computer. A force, proportional to the force exerted on the robot arm, is applied to the stylus of the SensAble® haptic device. The profile of the system is shown in Fig. 1. Its structure can be seen in Fig. 2 as well as the block diagram and the flow of information of the system.

* Corresponding authors: {cyang;mathmhb}@ieee.org

[1]Center for Robotics and Neural Systems, Plymouth University, Devon PL4 8AA, UK; [2]North Automatic Control Technology Institute, Taiyuan, 030006, P. R. China; [3] Key Laboratory of Autonomous System and Network Control, College of Automation Science and Engineering, South China University of Technology, Guangzhou, 510640, P. R. China; [4]Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, P. R. China; [5]State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing, 100081, P. R. China.

This work was supported in part by EPSRC grants EP/L026856/1 and EP/J004561/1; EU grant PIIFR-GA-2010-910078; and National Natural Science Foundation of China under grants 61473120 and 61473038.

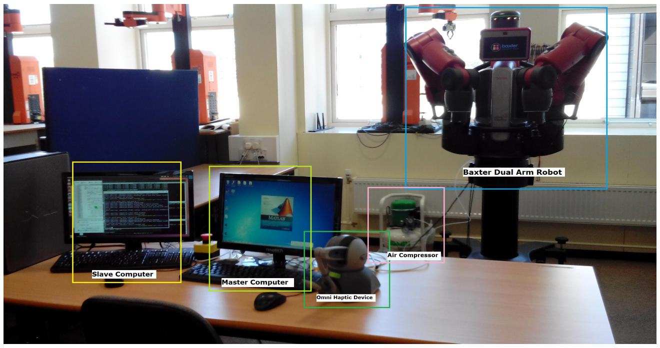


Fig. 1. Profile of the teleoperation system

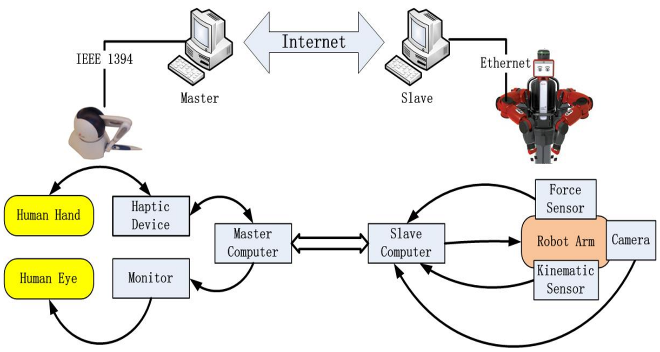


Fig. 2. Connection of the system and flow of information

A. Master Device

The SensAble Omni Haptic Device (SensAble® haptic Technologies) has 6 DOFs. The first three DOF's describe the position of the end-effector, while the last three form a gimbal and are related to its orientation. A stylus with two buttons is attached to the end-effector. The SensAble Omni has been very popular in the research area of teleoperation during the last years. Veras *et al* utilized the Omni device on the master side to teleoperate a Puma 560 robotic arm. They achieved assistance for daily activities of disabled people [10]. Chi and Zhang used the Omni haptic device as a master to control a virtual fixture in a tele-rehabilitation system [11]. The Omni could also be used to control a hydraulic machine, e.g., in [12] the Omni device was used to command the boom and stick segments of an excavator. Due to the popularity of the Omni device, the kinematics, including forward kinematics, inverse kinematics and the Jacobian, have been well studied in the literature [13], [14]. Mohammadi *et al* developed a simulation kit to support the Omni using the MATLAB/Simulink platform [15]. In this paper, a stable and high-performance 32/64bit driver has been designed. All control strategies have been programmed on the MATLAB/Simulink platform.

B. Slave Device

The Baxter® robot is formed by a torso on a movable pedestal and two 7-DOF arms installed on left/right arm mounts, respectively. Each arm has 7 rotational joints and 8 links in addition to a number of different interchangeable grippers, such as electric gripper or a vacuum cup, which can be installed at the end of each arm. A head-pan located on the top of the torso carries a screen and could rotate in the horizontal plane. We proposed a method to calculate the forward kinematics model and the DH (Denavit - Hartenberg) parameters of the Baxter robot[16], which has been adopted in this work.

III. Workspace Mapping

Operating a kinematic dissimilar teleoperating system, the manipulator of the slave is only able to work within the limits of the reachable workspace. Since the workspace and region boundaries of master and slave are often quite different, the evaluation of whether or not a given location is reachable is a fundamental issue. To resolve this problem, several mapping approaches have been developed. Those methods are able to map the master motion trajectories into a reachable workspace for the slave manipulator [17], [18], [19].

Analytical methods determine closed-form descriptions of the workspace boundaries. Unfortunately, these methods are usually complicated and complex, caused by nonlinear equations and matrix inversions, which are involved in the manipulator kinematics. Numerical methods, on the other hand, are known to be relatively simple and more flexible. Rastegar and Perel introduced the Monte Carlo random sampling numerical method to generate the workspace boundaries of some simple manipulators just using forward kinematics [20]. The method is fairly easy to apply and was chosen to create the workspace mapping model of this paper.

A. Forward Kinematics

Forward kinematics describe the relationship between the joint angles of the serial manipulator and the position and orientation of its end effector. The master of the system has a 6-DOF stylus and its kinematic model is described in Fig. 3.

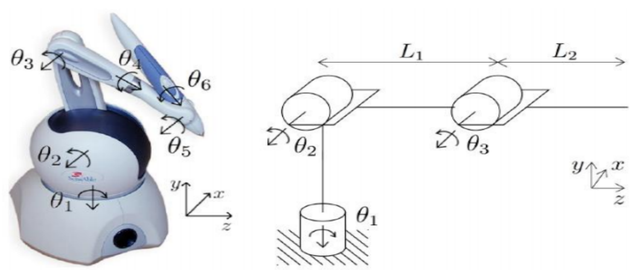


Fig. 3. Kinematic model of the Omni device

The DH parameters of the Omni device can be calculated as shown in Table I, where a_i are the link lengths, α_i the twist angles, d_i the link offsets, and θ_i the joint angles, which are variable. The link lengths of the Omni device are given

TABLE I
DH parameters of the Omni device

Link i	$\theta_i(anglelimit(deg))$	d_i	a_i	$\alpha_i(rad)$
1	$q_1(-60 \sim 60)$	0	0	$\frac{\pi}{2}$
2	$q_2(0 \sim 105)$	0	$L1$	0
3	$q_3(-100 \sim 100)$	0	0	$-\frac{\pi}{2}$
4	$q_4(-145 \sim 145)$	$-L2$	0	$\frac{\pi}{2}$
5	$q_5(-70 \sim 70)$	0	0	$-\frac{\pi}{2}$
6	$q_6(-145 \sim 145)$	0	0	$\frac{\pi}{2}$

by $L1 = L2 = 133.35mm$. Since the joints 4,5 and 6 form a spherical wrist, the link lengths and link offsets are set to zero.

The slave is a 7-DOF manipulator and its kinematic model is described in Fig. 4. The DH parameters of the Baxter arm

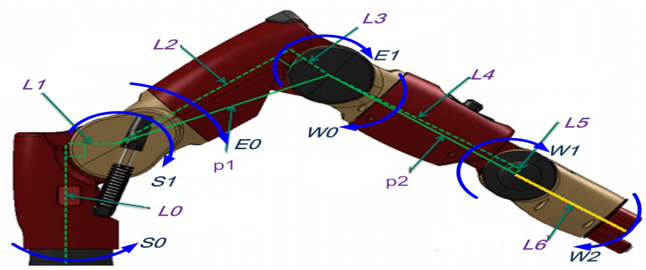


Fig. 4. Kinematic model of the Baxter arm

TABLE II
DH PARAMETERS OF THE BAXTER ARM

Link i	$\theta_i(anglelimit(deg))$	d_i	a_i	$\alpha_i(rad)$
1	$q_1(-97.5 \sim 97.5)$	$L0$	$L1$	$-\frac{\pi}{2}$
2	$q_2 + \frac{\pi}{2}(-123 \sim 60)$	0	0	$\frac{\pi}{2}$
3	$q_3(-175 \sim 175)$	$L2,$	$L3$	$-\frac{\pi}{2}$
4	$q_4(-2.865 \sim 150)$	0	0	$\frac{\pi}{2}$
5	$q_5(-175.27 \sim 175.27)$	$L4$	$L5$	$-\frac{\pi}{2}$
6	$q_6(-90 \sim 120)$	0	0	$\frac{\pi}{2}$
7	$q_7(-175.27 \sim 175.27)$	$L6$	0	0

can be calculated as shown in Table II. The link lengths of the Baxter arm are given by $L0 = 0.27m$, $L1 = 0.069m$, $L2 = 0.364m$, $L3 = 0.069m$, $L4 = 0.375m$, $L5 = 0.01m$, and $L6 = 0.28m$.

Using the DH notation method for modelling the manipulator, the links of the manipulator are numbered from 1, addressing the base link, to n, which is the outermost link or hand. A coordinate system is attached to each link to describe the relative arrangements among the various links. The coordinate system, attached to the ith link, has the indices i as well. The 4×4 transformation matrix, which relates the $i + 1th$ coordinate system to the ith coordinate system is

$$^{i-1}A_i(\theta_i) = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ c\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where $s\theta_i = sin\theta_i$, $c\theta_i = cos\theta_i$ and θ_i is the ith joint rotation angle. Furthermore, α_i is the twist angle, a_i and d_i are the length of link $i+1$ and its offset distance from joint i [21]. Referring to the DH table and using equation (1), the homogeneous transformation matrices can be calculated through matrices multiplication as the following:

$$^0A_n = {}^0A_1 {}^1A_2 \cdots {}^{n-1}A_n \quad (2)$$

The frame axis direction of the Omni device is different from the frame direction of the Baxter, as illustrated in Fig. 5. Therefore, the Cartesian coordinates of Omni,

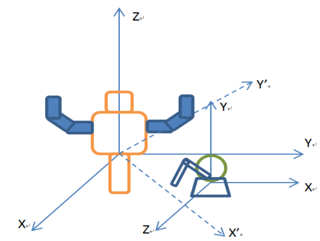


Fig. 5. Frame axis direction of the Omni and the Baxter.

$[x_m{}' y_m{}' z_m{}']^T$, needs to be modified according to the equation below:

$$A_o{}' = R_z(\frac{\pi}{2})R_x(\frac{\pi}{2})A_o R_y(\frac{\pi}{2})R_z(\frac{\pi}{2}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3)$$

where A_o represents the transform matrix of the Omni device and $A_o{}'$ is the corresponding modified matrix.

The forward kinematics specify the relation between Cartesian space and joint space and can be represented by the following equation

$$\overline{X_0} = ({}^0A_n)^T \overline{X_n} \quad (4)$$

Assuming that all joint angles are known, the multiplication of the transformation matrices, combined with equation (4), results in the forward kinematics transformation from the base to the end effector of the manipulator.

B. Generating Approximate Workspace

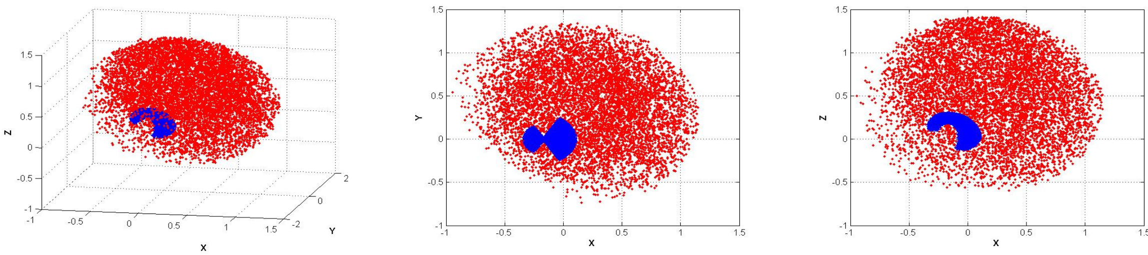
The Monte Carlo method of random sampling is applied to the joint space of the manipulator to approximate the workspace. Table I and table II give the joint rotation limits of the master and slave workspaces. Bart Milne *et al* used a distance loop in all joint spaces to generate the contour, but it tended to be time consuming[6]. Therefore, in this work, homogeneous radial distribution is applied to generated 8000 points in the joint spaces of master and slave separately. Furthermore, by using the forward kinematics model of equation (4), the workspace was generated as shown in Fig. 6.

C. Workspace Mapping

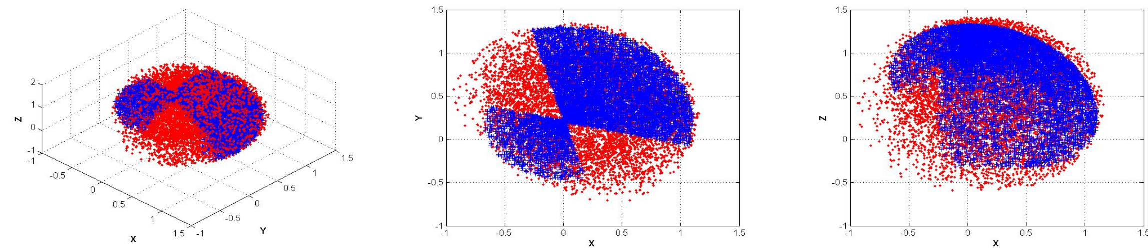
As emphasized above, the workspace mapping is done to let the workspaces of master and slave overlap each other as much as possible, improving the manoeuvrability of the slave. The point cloud matching method is utilized, since it is convenient by considering the position of the end effectors instead of the structures of the different devices. The mapping process can be described as the following equation

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} cos\delta & -sin\delta & 0 \\ sin\delta & cos\delta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \left(\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \right) \quad (5)$$

where $[x_s y_s z_s]^T$, $[x_m y_m z_m]^T$ are the Cartesian coordinates of the end effectors of Baxter and Omni respectively, δ is the revolution angle about the Z-axis of the Baxter base frame and $[S_x S_y S_z]^T$ and $[T_x T_y T_z]^T$ are the scaling factors and translations about the X, Y and Z axis. For the left arm of



(a) Contour plot before workspace mapping



(b) Contour plot after workspace mapping

Fig. 6. Workspace mapping. The red pointed cloud represents the workspace of the slave, while the blue pointed cloud represents the workspace of the master.

the Baxter robot, the mapping parameters of equation (5) are given by

$$\delta = \frac{\pi}{4}, \quad \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} 0.0041 \\ 0.0040 \\ 0.0041 \end{bmatrix}, \quad \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} 0.701 \\ 0.210 \\ 0.129 \end{bmatrix}$$

Equation (5) only regards the revolution about the Z-axis. The Omni joystick is placed on a horizontal platform and, as a consequence, the Z-axis of the master is perpendicular to the horizontal plane. Likewise, the Baxter robot is adjusted carefully to make sure that the Z-axis is upward vertical and also perpendicular to the horizontal plane. Thereby, the Z-axis of the master is parallel to the Z-axis of the slave. The revolution about the X- and Y-axis are neglected due to the applied human-in-loop control approach. Assuming that the X- and Y-axis of master and slave are aligned approximately, the control of the slave by using the accurate coordinate values directly becomes needles. The operator can adjust the arm of the robot according to the posture depicted on the master computer's screen.

IV. Command Strategies

To control the slave, the position-position scheme has been applied. In [22], Farkhatdinov *et al* approached a position-position command strategy to steer a mobile robot with two wheels through a corridor including prescribed stopping points along the way. A Phantom Premium 1.5A was used as the master. Furthermore, this strategy was used in [23] and input saturation compensation has been investigated as well. In this paper, the numerical inverse kinematics method is used to calculate the joint configuration. Afterwards, the joint angles are sent to the slave, realizing the control of the end-effector.

A. Joint Space Position-Position Control

To implement the position-position control strategy, one approach is to map the joint angles of the master to the slave directly, applying teleoperation in the joint space. Since the Baxter arm is a redundant manipulator and has 7 joints, while the Omni device only has 6 joints, a certain joint of the slave has to be restricted for this method. Considering the similarity of the geometric structure of master and slave, the third joint is fixed in the direct angle mapping method. The angle of the joint *E0* is set to zero. Matching the other joints one by one, the slave can move accordingly to the master. The described 6DOF-6DOF direct angle mapping method represents a convenient choice for the position-position control strategy. Ahn *et al* proved that by restricting a 6-DOF master and 7-DOF slave to just 2 DOFs. Afterwards, the two axes of master and slave could be matched directly [23].

B. Task Space Position-Position Control

To take advantage of the 7-DOF slave's redundant manipulator, a inverse kinematics algorithm needs to be developed to implement the Cartesian position transformation from master to slave. In this paper, the CLIK method is applied.

If the DH parameters are known, the relation between the task vector $\overline{x}$ and the joint vector $\overline{q}$ will be represented [21] by

$$\overline{x} = f(\overline{q}) \tag{6}$$

where f is a continuous nonlinear function of known structure and noted parameters. The fundamental task of teleoperation is to solve the inverse kinematics problem, i.e, solving equation (6) regarding $\overline{q}$. The most popular approach for solving this problem is to find a closed-form analytical solution for equation (6). However, this method may be resolvable only for simple geometries. Differentiating equation (6) with respect to time yields to the mapping of the joint velocity vector $\dot{\overline{q}}$ into the end-effector task velocity vector $\dot{\overline{x}}$,

$$\dot{\overline{x}} = J(\overline{q})\dot{\overline{q}} \tag{7}$$

where $J(\overline{q}) = \frac{\partial f}{\partial q}$ is Jacobian matrix [24]. Inserting the Jacobian matrix in equation (7), $\dot{q}$ can be written as

$$\dot{\overline{q}} = J^{+}(\overline{q})\dot{\overline{x}} \tag{8}$$

where $J^{+}(\overline{q})$ is the pseudo-inverse of the Jacobian matrix and can be computed as $J^{+} = J^{T}(JJ^{T})^{-1}$. The joint displacements $\overline{q}$ are then obtained by integrating equation (8) over time.

However, the method using equation (8) is inherently *open-loop* and causes numerical drifts in the task space unavoidably. In order to overcome this drawback, the CLIK (*closed loop* inverse kinematics) algorithm is applied. The task space vector $\dot{\overline{x}}$ is replaced by $\dot{\overline{x}} = Ke$ in (8), where $e = \overline{x_d} - \overline{x}$ describes the error between the desired task trajectory $\overline{x_d}$ and the actual task trajectory $\overline{x}$. It can be computed from the current joint variables via the forward kinematics equation (6). K is a positive definite matrix, that shapes the error convergence [25], [26]. Considering the computational burden, the pseudo-inverse of the Jacobian in (8) can be replaced by the transposed Jacobian. Then equation (8) can be written as

$$\dot{\overline{q}} = KJ^{T}(\overline{q})e \tag{9}$$

This solution may avoid the typical numerical instabilities,

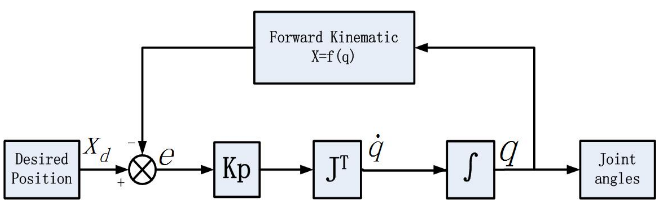


Fig. 7. The block diagram of the CLIK algorithm

which occur at kinematic singularities, since no pseudo-inverse of the Jacobian matrix is required. The details and the steady state proof of this solution are shown in [27], [28], [29]. The block diagram of the CLIK algorithm is shown in Fig. 7.

C. Haptic Rendering Algorithm

The command strategy is based on the position-position scheme. It is difficult for the human operator to estimate the position error between the master and the slave just through the simulation of robot motion on the screen. Therefore, a haptic rendering algorithm, based on error information, should be added to the system. In the current setting, force feedback is proportional to the amplitude of position error, using a simple rendering algorithm based on Hook's Law. This haptic algorithm acts as an indicator. Once the error is growing, the force feedback will increase proportional to it, which in turn aids the operator to gain situation awareness and improve the overall accuracy of the performed motions.

The absolute value of the feedback force can be calculated as

$$|f| = K_f\sqrt{(x_s - x_d)^2 + (y_s - y_d)^2 + (z_s - z_d)^2} \tag{10}$$

where K_f is the feedback force scaling factor, $(x_s y_s z_s)$ is the actual Cartesian position of the slave's end effector and $(x_d y_d z_d)$ is the desired position.

V. Experiment

To verify the effectiveness of the proposed teleoperation system, two experiments have been carried out[1].

A. Free Space Moving

Free space movements are used to test the position-position command performance. The operator holds the Omni stylus firmly and moves slowly in its workspace. First a move in translational direction and then a turn of the stylus into the desired orientation are performed. These two actions can be used to check the translational and rotational capability of the system, respectively. Fig. 8 shows the results of the experiment. The blue curve represents the trajectory of the master. The green one represents the trajectory of the end-effector of the slave. From the results it can be concluded, that the position-position teleoperational control algorithm works well. The maximum position error between master and slave is not bigger than $0.1m$. Considering the error of workspace mapping and the fact that the position controller follows a simple PD strategy, the results are acceptable.

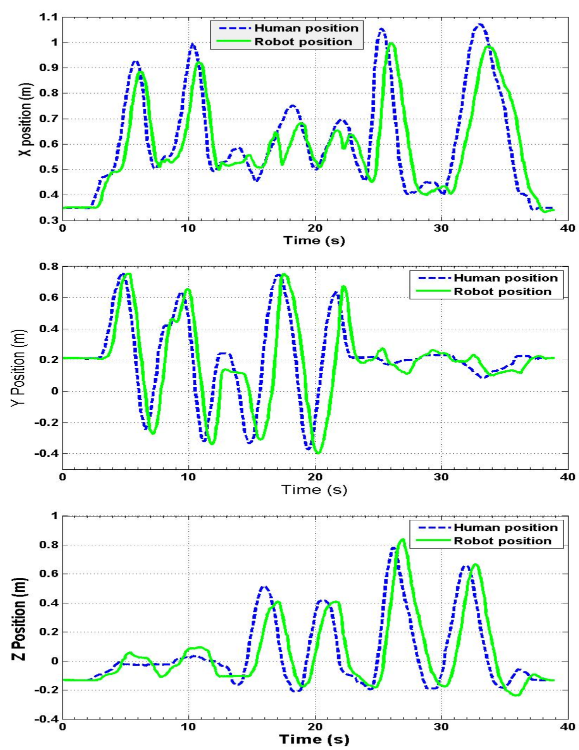


Fig. 8. Robot moving in the free space

B. Haptic Feedback Test

Haptic feedback features the system through force feedback, sent from the slave manipulator to the master. This enables the operator to sense the remote robot and manipulate more naturally. While the human moves the stylus of the slave, the feedback force is applied to it continuously. This experiment aims at checking the force feedback, received

[1]An illustrative video of the experiment can be viewed at: https://www.youtube.com/watch?v=YDvIAmc2thA or http://v.youku.com/v_show/id_XNzYzMzY4NTUy.html

from the manipulator and exerted on the stylus, while moving the stylus slowly. The results are shown in Fig. 9.

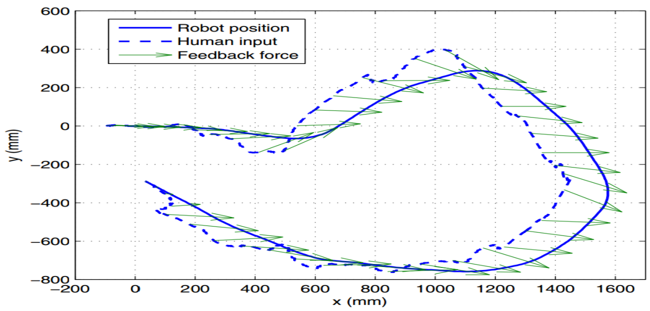


Fig. 9. The feedback force while the manipulator is moving

The dotted line represents the trajectory of the stylus and the solid line represents the actual trajectory of the manipulator. The arrows represent direction and magnitude of the feedbacked force. The feedback force provides direct ratio of the position error and points towards the manipulator. This feedback force is provided by the control strategy given in equation (10), which measures the error between the actual position of manipulator and the desired position.

VI. Conclusions

This paper presented a position-position control strategy with force feedback for teleoperating system, in which the slave robot arm moves according to the motion of the master. A workspace mapping method was proposed, facilitating the opportunity to match the workspace boundaries of the master and the slave. The CLIK algorithm ensured the accurate and flexible position transformation between master and slave. A haptic rendering algorithm was designed for the haptic feedback interaction. With the help of the described system, the user is able to sense the position error as well as to observe the simulated action and video displayed in real time on the monitor. The features of the described system can help people to interact with the environment of remotely controlled robots in a haptic manner.

References

[1] H. I. Krebs, N. Hogan, M. L. Aisen, and B. T. Volpe, "Robot-aided neurorehabilitation," *Rehabilitation Engineering, IEEE Transactions on*, vol. 6, no. 1, pp. 75–87, 1998.

[2] S. Guo, G. Song, and Z. Song, "development of a self-assisted rehabilitation system for the upper limbs based on virtual reality," *Mechatronics*, 2007.

[3] C. Li, H. Ma, C. Yang, and M. Fu, "Teleoperation of a virtual icub robot under framework of parallel system via hand gesture recognition," in *Proceedings of the 2014 IEEE World Congress on Computational Intelligence*. Beijing: WCCI, Jul 6-11 2014.

[4] K. Inoue, T. Tanikawa, and T. Arai, "Micro hand with two rotational fingers and manipulation of small objects by teleoperation," in *Micro-NanoMechatronics and Human Science, 2008. MHS 2008. International Symposium on*. IEEE, 2008, pp. 97–102.

[5] P. Renon, C. Yang, H. Ma, and M. Fu, "Haptic interaction between human and virtual robot with chai3d and icub simulator," in *Proceedings of the 32nd Chinese Control Conference (CCC2013)*. Xi'an: CCC, 2013.

[6] G. C. Burdea, C. Burdea, and C. Burdea, *Force and touch feedback for virtual reality*. Wiley New York, 1996.

[7] K. B. Shimoga, "A survey of perceptual feedback issues in dexterous telemanipulation. ii. finger touch feedback," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*. IEEE, 1993, pp. 271–279.

[8] R. J. Stone, "Haptic feedback: A brief history from telepresence to virtual reality," in *Haptic Human-Computer Interaction*. Springer, 2001, pp. 1–16.

[9] H. Wang and X. P. Liu, "Design of a novel mobile assistive robot with haptic interaction," in *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference on*. IEEE, 2012, pp. 115–120.

[10] E. Veras, K. Khokar, R. Alqasemi, and R. Dubey, "Scaled telerobotic control of a manipulator in real time with laser assistance for adl tasks," *Journal of the Franklin Institute*, vol. 349, no. 7, pp. 2268–2280, 2012.

[11] P. Chi and D. Zhang, "Virtual fixture guidance for robot assisted teleoperation."

[12] H. Hayn and D. Schwarzmann, "Control concept for a hydraulic mobile machine using a haptic operating device," in *Advances in Computer-Human Interactions, 2009. ACHI'09. Second International Conferences on*. IEEE, 2009, pp. 348–353.

[13] T. Sansanayuth, I. Nilkhamhang, and K. Tungpimolrat, "Teleoperation with inverse dynamics control for phantom omni haptic device," in *SICE Annual Conference (SICE), 2012 Proceedings of*. IEEE, 2012, pp. 2121–2126.

[14] A. J. Silva, O. A. D. Ramirez, V. P. Vega, and J. P. O. Oliver, "Phantom omni haptic device: Kinematic and manipulability," in *Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA'09*. IEEE, 2009, pp. 193–198.

[15] A. Mohammadi, M. Tavakoli, and A. Jazayeri, "Phansim: A simulink toolkit for the sensable phantom haptic devices," in *23rd Canadian Congress of Applied Mechanics, Vancouver, BC, Canada*, 2011, pp. 787–790.

[16] Z. Ju, C. Yang, and H. Ma, "Kinematics modeling and experiment verification for baxter robot," in *Proceedings of the 33rd Chinese Control Conference (CCC2014)*. Nanjing: CCC, 2014.

[17] H. Wang, K. H. Low, F. Gong, and M. Y. Wang, "A virtual circle method for kinematic mapping human hand to a non-anthropomorphic robot," in *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, vol. 2. IEEE, 2004, pp. 1297–1302.

[18] R. V. Dubey, S. Everett, N. Pernalete, and K. A. Manocha, "Teleoperation assistance through variable velocity mapping," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 5, pp. 761–766, 2001.

[19] N. Pernalete, W. Yu, R. Dubey, and W. Moreno, "Development of a robotic haptic interface to assist the performance of vocational tasks by people with disabilities," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2. IEEE, 2002, pp. 1269–1274.

[20] J. Rastegar and D. Perel, "Generation of manipulator workspace boundary geometry using the monte carlo method and interactive computer graphics," *Journal of mechanical design*, vol. 112, no. 3, pp. 452–454, 1990.

[21] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011, vol. 73.

[22] I. Farkhatdinov and J.-H. Ryu, "Hybrid position-position and position-speed command strategy for the bilateral teleoperation of a mobile robot," in *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*. IEEE, 2007, pp. 2442–2447.

[23] S. H. Ahn, B. S. Park, and J. S. Yoon, "A teleoperation position control for 2-dof manipulators with control input saturation," in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, vol. 3. IEEE, 2001, pp. 1520–1525.

[24] J. J. Craig, *Introduction to robotics*. Addison-Wesley Reading, MA, 1989, vol. 7.

[25] L. Sciavicco and B. Siciliano, "A dynamic solution to the inverse kinematic problem for redundant manipulators," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 1081–1087.

[26] Y. T. Tsai and D. E. Orin, "A strictly convergent real-time solution for inverse kinematics of robot manipulators," *Journal of robotic systems*, vol. 4, no. 4, pp. 477–501, 1987.

[27] L. Sciavicco and B. Siciliano, "Solving the inverse kinematic problem for robotic manipulators," in *RoManSy 6*. Springer, 1987, pp. 107–114.

[28] J.-J. Slotine and D. Yoerger, "A rule-based inverse kinematics algorithm for redundant manipulators." *Int. J. Robotics Autom.*, vol. 2, no. 2, pp. 86–89, 1987.

[29] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *Robotics and Automation, IEEE Journal of*, vol. 4, no. 4, pp. 403–410, 1988.