Front. Electr. Electron. Eng. China 2007, 2(3): 288–292
DOI 10.1007/s11460-007-0053-4

RESEARCH ARTICLE

LIU Wei, SONG Aiguo, LI Huijun

Force-reflecting teleoperation of robots based on on-line correction of a virtual model

Abstract Virtual reality is an effective method to eliminate the influence of time delay. However, it depends on the precision of the virtual model. In this paper, we introduce a method that corrects the virtual model on-line to establish a more precise model. The geometric errors of the virtual model were corrected on-line by overlapping the graphics over the images and also by syncretizing the position and force information from the remote. Then the sliding average least squares (SALS) method was adopted to determine the mass, damp, and stiffness of the remote environment and use this information to amend the dynamic model of the environment. Experimental results demonstrate that the on-line correction method we proposed can effectively reduce the impact caused by time delay, and improve the operational performance of the teleoperation system.

Keywords force-reflecting, teleoperation, time delay, virtual model, on-line correction

1 Introduction

In the force-reflecting teleoperation system, the position information is sent from the master side to the slave side, and the force information is then sent to the master side. Thus, the control loop is formed. If there were no time delay in the loop, the system could be stable. However, because the force-reflecting teleoperation system is often used in long-distance situations, the time delay between the operator and the remote robot has been the main problem that influences the remote task. Many methods have been proposed to solve this problem, which can be mainly classified into two kinds: one is concentrating on the control laws, including the passivity and scattering theory [1]; another is to use virtual reality tools. By building a virtual model of the remote environment, the operator can interact with the virtual model without time delay. The virtual model predicts the motion of the remote environment, and thus overcoming the time delay problem. Because the environment is unknown or partly known, the errors are inevitable in the predictive virtual model, which will intensify the increase of time delay and result in the unstableness of the system. The existing control laws can only be applied to short time delay (less than 2–3 s) [2–4]. In 1997, Tsumaki proposed a teleoperation system that is robust against modeling errors [5]. In 1999, Burdea concluded the use of virtual reality in force-reflecting teleoperation with time delay and indicated that it had two disadvantages: first, the virtual models of the slave, remote environment, and task were oversimplified; second, the approach will fail in unstructured environments which cannot be modeled accurately [6]. The predictive model of the slave has been set up. With the information from the slave, the model is corrected on-line to make the virtual model more similar to the real environment. Therefore, the interaction with the virtual model can be more consistent with the real environment.

2 Influence of errors

The virtual predictive models of the slave arm and environment are first set up. Then, the operator commands the model of the slave arm to interact with the virtual environment through the master arm, and the contact information is also sent to the operator through the controller in the master. At the same time, the command is sent to the slave to control the slave arm (interact with the environment) with time delay. If the virtual models are consistent with the actual environment, the operator can control the virtual arm to interact with the virtual environment, which is equivalent to controlling the slave arm to interact with the remote environment. Then the influence caused by time delay can be eliminated completely. The virtual models can give the operator the accurate force reflecting with the time delay between the master and the slave only when they are set up precisely. However, the errors in the virtual predictive model may result in the errors in

---

Translated from *Journal of Southeast University* (*Natural Science*), 2006, 36(2): 242−246 [译自: 东南大学学报(自然科学版)]

LIU Wei, SONG Aiguo (✉), LI Huijun
Department of Instrument Science and Engineering, Southeast University, Nanjing 210096, China
E-mail: a.g.song@seu.edu.cn

the reflecting force. There are two types of errors in the virtual model—geometrical errors (position errors) and dynamic errors (force and motion errors). To get the accurate reflecting force, we must eliminate the influence of the virtual model errors.

3 Construction and correction of the model

In remote control, the time delay between the master and the slave is often quite long, which can be 2−10 s. The long time delay seriously damages the stability and transparency of the systems. The virtual predictive model is an effectual way to solve the time delay problem, but it depends on the accuracy of the model. The applications of force-reflecting teleoperation are commonly in the unknown or half known environment. It is difficult to construct an accurate geometrical and dynamic model, so there are always some errors in the parameters of the model. Low accuracy causes the low efficiency of the systems, which can prevent the use of virtual reality tools. As to unifying the virtual model to the actual environment, the on-line correction is entailed.

The diagram of the virtual model construction and its on-line correction is shown in Fig. 1. It can be divided into the following parts:

1) data acquisition in the slave: the charged coupled device (CCD) image with position and force information in the slave is collected with the sensors, and then transferred to the master;

2) virtual model construction: construct the line frame model of the slave according to the collecting information; construct the dynamic model according to the identification consequence;

3) virtual model correction: update the position of the virtual model according to the master and slave position information, and update the dynamic parameters of the model according to the information from the sensors and the effect of the following force;

4) graphics and image information fusion: lay the line frame model on the image of the task environment and show them together.

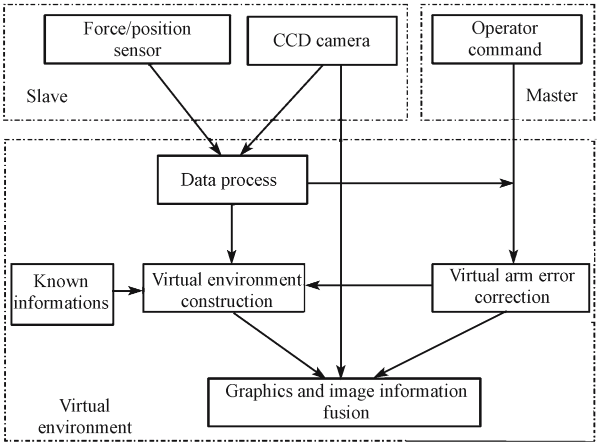

Fig. 1 The on-line correction of virtual model

3.1 Geometrical error correction

To the operator, the image from the remote camera already contains the communication time delay. For the real-time visual information, the virtual predictive models are utilized. The operator can control the virtual model just like controls slave arm, and real-time visual feedback can be implemented. The real-time virtual graphics are also overlaid on the static task image with communication time delay, and through this enhanced reality, we can easily "detect" the geometrical errors between virtual predictive model and real slave arm and directly correct it.

Because the errors are accumulated with the motion, the position offsets exist between the virtual arm and real arm. Thus, we need to eliminate the offsets according to the position of the virtual arm and real arm. As to the one degree of freedom (DOF) system, we can use the following equation to correct the position of virtual arm on-line, that is, the geometrical errors correction.

$$x_{\mathrm{v}}(t) = x_{\mathrm{m}}(t) + \big(x_{\mathrm{m}}(t-2\tau) - x_{\mathrm{s}}(t-\tau)\big) \tag{1}$$

Here, x_{v} is the position of the virtual arm. x_{m} is the position of the master arm, x_{s} is the position of the slave arm, and τ is the one way communication time delay.

First, the geometrical model of the environment is set up through camera information. Then, the geometrical errors are corrected through the information fusion of position and force. After the initialization of the virtual environment, the operator controls the motion of the master arm and commands the virtual arm and slave arm to perform the same task. The force and position sensor information in the slave are continuously transmitted to the master. If the force sensor detects a step signal, it means that the slave arm starts to contact the environment, and the position of the slave arm in this moment representing the environment initial position. The initialized virtual environment is then checked out in this way.

3.2 Dynamic error correction

In the force-reflecting teleoperation system, the virtual model should provide force feedback. The dynamic model is very important in calculating the virtual force. It is the precondition for virtual force reflecting and the time delay problem overcoming. Because the environment dynamic changes with the position, its parameter is quite difficult to identify. Erickson in his paper compared many kinds of algorithms for the identification of contact stiffness and damping during robot constrained motion, especially the dynamic modeling and simulation of robotic assembly operations in space [7]. Diolaiti also employed the Hunt-Crossley model to estimate on-line the dynamic characteristics of the environment, but the measurement of force, position and velocity is necessary [8]. Lonnie combined a conventional multi-input, multi-output recursive least squares (MIMO-RLS) system identi-fication with a discretized representation of the remote environment, and identified in real-time the remote environment. The accuracy of this algorithm is highly dependent on the computing and storage capability. If the discretization is too specific, real-time identification will be difficult to acquire because of the limit of computing and storage capability. While conversely, the dynamic parameters of the environment could not be clearly clarified [9]. All the above-mentioned adopt the nonlinear model to describe the dynamic parameters of the environment and the contact model while needing a great deal of computation, which are not fit for the real-time identification and correction. In this paper, an algorithm that can rapidly estimate and correct the environment is proposed.

The dynamic characteristics of the environment are commonly described as a mass-damp-spring model. It means that the structure of the environment model is known. Our work is to identify the parameters of the environment dynamics. Based on this idea, we adopt sliding average least squares (SALS) method to make out the mass, damp and stiffness of the remote environment in real time, and build the dynamic model of the remote environment. First, we should figure out the relationship between the force and the position in restrained motion mode. Second, the parameters of environment dynamics are confirmed and corrected on-line with the information from the remote.

The dynamic model of the environment which interacts with the slave is as below

$$f_{\mathrm{e}} = m_{\mathrm{e}}\ddot{x}_{\mathrm{e}} + b_{\mathrm{e}}\dot{x}_{\mathrm{e}} + k_{\mathrm{e}}x_{\mathrm{e}} \tag{2}$$

where f_{e} is the contact force between slave arm and environment, x_{e} is the position of environment, m_{e}, b_{e}, k_{e} denote the mass, damp, and stiffness of the environment, respectively. The control structure is shown in Fig. 2. The dynamic characteristics of the environment are identified in real time and the on-line correction of the virtual model is shown. The identification is finished in the slave to lessen the burden of the computation in the master to promote the accuracy of identification.

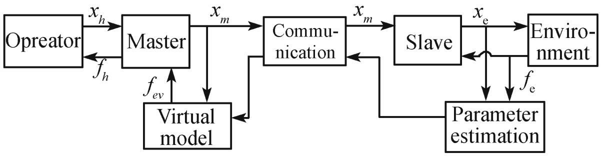

Fig. 2 Environment dynamic estimation

x_h, f_h denote the position command and contact force of operator, respectively, x_{m} *is* the position of the master. $\hat{m}_{\mathrm{e}}$, $\hat{b}_{\mathrm{e}}$, $\hat{k}_{\mathrm{e}}$ are the estimated value of m_{e}, b_{e}, k_{e}, respectively, then

$$\hat{f}_{\mathrm{e}} = \hat{m}_{\mathrm{e}}\ddot{x}_{\mathrm{e}} + \hat{b}_{\mathrm{e}}\dot{x}_{\mathrm{e}} + \hat{k}_{\mathrm{e}}x_{\mathrm{e}} \tag{3}$$

According to the least squares criterion

$$E = \sum_{i=1}^{N}\left[f_{\mathrm{e}}(i) - \hat{f}_{\mathrm{e}}(i)\right]^2 \tag{4}$$

$$\frac{\partial E}{\partial \hat{m}_{\mathrm{e}}} = 0\,,\ \frac{\partial E}{\partial \hat{b}_{\mathrm{e}}} = 0\,,\frac{\partial E}{\partial \hat{k}_{\mathrm{e}}} = 0 \tag{5}$$

Then

$$\begin{bmatrix}\hat{m}_{\mathrm{e}}\\ \hat{b}_{\mathrm{e}}\\ \hat{k}_{\mathrm{e}}\end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N}\ddot{x}_{\mathrm{e}}^2(i) & \sum_{i=1}^{N}\ddot{x}_{\mathrm{e}}(i)\dot{x}_{\mathrm{e}}(i) & \sum_{i=1}^{N}\ddot{x}_{\mathrm{e}}(i)x_{\mathrm{e}}(i)\\ \sum_{i=1}^{N}\dot{x}_{\mathrm{e}}(i)\ddot{x}_{\mathrm{e}}(i) & \sum_{i=1}^{N}\dot{x}_{\mathrm{e}}^2(i) & \sum_{i=1}^{N}\dot{x}_{\mathrm{e}}(i)x_{\mathrm{e}}(i)\\ \sum_{i=1}^{N}x_{\mathrm{e}}(i)\ddot{x}_{\mathrm{e}}(i) & \sum_{i=1}^{N}x_{\mathrm{e}}(i)\dot{x}_{\mathrm{e}}(i) & \sum_{i=1}^{N}x_{\mathrm{e}}^2(i) \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{i=1}^{N}\ddot{x}_{\mathrm{e}}(i)f_{\mathrm{e}}(i)\\ \sum_{i=1}^{N}\dot{x}_{\mathrm{e}}(i)f_{\mathrm{e}}(i)\\ \sum_{i=1}^{N}x_{\mathrm{e}}(i)f_{\mathrm{e}}(i) \end{bmatrix} \tag{6}$$

N is the sampling number of characteristics estimation. For the on-line estimation, we adopt the sliding average method

$$\begin{bmatrix}\hat{m}_{\mathrm{e}}\\ \hat{b}_{\mathrm{e}}\\ \hat{k}_{\mathrm{e}}\end{bmatrix} = \begin{bmatrix} \sum_{i=t-N+1}^{t}\ddot{x}_{\mathrm{e}}^2(i) & \sum_{i=t-N+1}^{t}\ddot{x}_{\mathrm{e}}(i)\dot{x}_{\mathrm{e}}(i) & \sum_{i=t-N+1}^{t}\ddot{x}_{\mathrm{e}}(i)x_{\mathrm{e}}(i)\\ \sum_{i=t-N+1}^{t}\dot{x}_{\mathrm{e}}(i)\ddot{x}_{\mathrm{e}}(i) & \sum_{i=t-N+1}^{t}\dot{x}_{\mathrm{e}}^2(i) & \sum_{i=t-N+1}^{t}\dot{x}_{\mathrm{e}}(i)x_{\mathrm{e}}(i)\\ \sum_{i=t-N+1}^{t}x_{\mathrm{e}}(i)\ddot{x}_{\mathrm{e}}(i) & \sum_{i=t-N+1}^{t}x_{\mathrm{e}}(i)\dot{x}_{\mathrm{e}}(i) & \sum_{i=t-N+1}^{t}x_{\mathrm{e}}^2(i) \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{i=t-N+1}^{t}\ddot{x}_{\mathrm{e}}(i)f_{\mathrm{e}}(i)\\ \sum_{i=t-N+1}^{t}\dot{x}_{\mathrm{e}}(i)f_{\mathrm{e}}(i)\\ \sum_{i=t-N+1}^{t}x_{\mathrm{e}}(i)f_{\mathrm{e}}(i) \end{bmatrix} \tag{7}$$

Rewrite it in matrix form

$$[\hat{\boldsymbol{Z}}_{\mathrm{e}}(t)] = [\boldsymbol{A}(t)]^{-1}[\boldsymbol{C}(t)]; \quad t \geqslant N \tag{8}$$

The element of matrix $\boldsymbol{A}(t)$, $\boldsymbol{C}(t)$ can be solved in the sliding average method.

$$\begin{aligned} a_{i,j}(t+1) &= \sum_{k=t-N+2}^{t+1} x_{\mathrm{e}}^{(3-i)}(k)x_{\mathrm{e}}^{(3-j)}(k)\\ &= a_{i,j}(t) + x_{\mathrm{e}}^{(3-i)}(t+1)x_{\mathrm{e}}^{(3-j)}(t+1)\\ &\quad - x_{\mathrm{e}}^{(3-i)}(t-N+1)x_{\mathrm{e}}^{(3-j)}(t-N+1)\\ &\quad i = 1,2,3; j = 1,2,3 \end{aligned} \tag{9}$$

$$\begin{aligned} c_i(t+1) &= \sum_{k=t-N+2}^{t+1} x_{\mathrm{e}}^{(3-i)}(k)f_{\mathrm{e}}(k)\\ &= c_i(t) + x_{\mathrm{e}}^{(3-i)}(t+1)f_{\mathrm{e}}(t+1)\\ &\quad - x_{\mathrm{e}}^{(3-i)}(t-N+1)f_{\mathrm{e}}(t-N+1);\\ &\quad i = 1,2,3 \end{aligned} \tag{10}$$

Using the corrected parameters $\hat{m}_e$, $\hat{b}_e$, $\hat{k}_e$, we can update on-line the parameters m_e, b_e, k_e. Then, we can calculate the virtual force reflecting from the virtual environment.

$$f_v = m_v\ddot{x}_v + b_v\dot{x}_v + k_v x_v \tag{11}$$

where x_v is the position of the virtual environment, and m_v, b_v, k_v are respectively the mass, damp, and stiffness of the environment.

4 Experiment results

We build a one-DOF force-reflecting teleoperation system based on the virtual predictive model, which is shown in Fig. 3. The system includes the master arm, the slave arm, two direct-current torque motors, two position and force sensors, two sets of data acquisition and motor drive systems, plus image acquisition and processing systems, and the virtual predictive model. The master and slave communicate with each other through the internet. The position and force information in the master are sent on one hand to the virtual model, which are used to produce the interaction information, and on the other hand to the slave, which are used to control the motion of the slave. The position and force information, which are transmitted from the slave to the master, are used in error compensation of the virtual model.

4.1 Geometrical error correction

We construct the virtual model from the image of the camera. Because the view of the camera is two- dimensional graphics, the virtual model is also in two dimensions. We set up the virtual model in line frame for the convenience of comparing the graphics with the image from the camera.

We use the open graphics library (OpenGL) to construct our one-DOF virtual arm and the virtual environment. The virtual model is set up in line frame for the sake of easier observation and comparison. The environment object is represented by a spring-mass model. The position of the environment object is unknown, and is identified in the visual system. The object is drawn in black line frame. The character of the virtual arm is the same as the slave arm and controlled by the operator in the same control law with the slave arm. Figure 4 shows the overlaid result after the correction. The background is the camera image, and the black line frame is the virtual model.

4.2 Dynamic error correction

We set the one-way communication time delay as 15 s through the software. The total communication time delay is 30 s (including the forward and backward time delay). The sampling frequency of the system is 100 Hz. In order to confirm the consistency between the virtual geometrical and dynamic model, we observed not only the information of the slave and master arm, but also the changes in the environment. The following results of position and force are shown in Fig. 5. Figure 6 shows the geometrical changes of the environment.

We also cut off a piece of data of the virtual force and actual feedback force of the slave for comparison. The data are smoothed, and the errors between the two kinds of forces are calculated, as shown in Fig. 7. The broken line represents the virtual force, and the real line is the actual force.

The errors between the forces become bigger when the slave arm starts to contact and to leave the environment object, which is due to the non-linear characteristic of the system in this moment.

5 Conclusion

Virtual reality is an effective method to solve the time delay problem in teleoperation. In this paper, the study is concentrated on how to improve the accuracy of the virtual predictive model. The geometrical and dynamic errors of the virtual predictive model are corrected on-line. It can be seen from the experiment results that the position and force of the virtual model are quite consistent with the slave arm with the real time correction. The on-line correction methods proposed in the article ensure the consistency between the force and position.

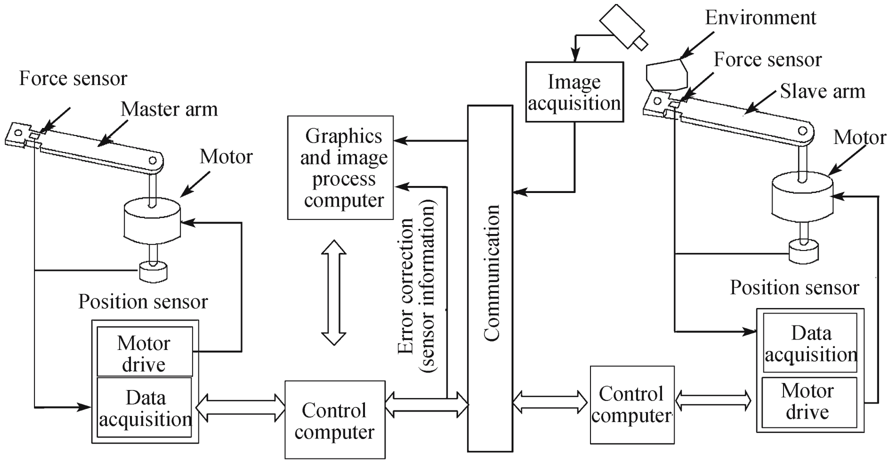

Fig. 3 Experiment system

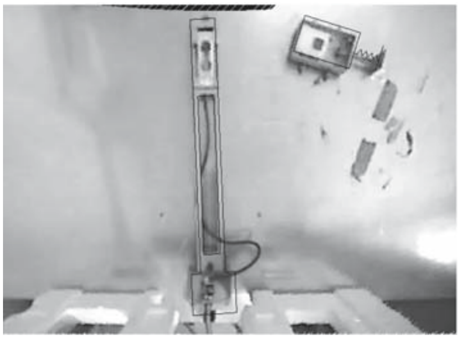

Fig. 4 The graphics and image overlay

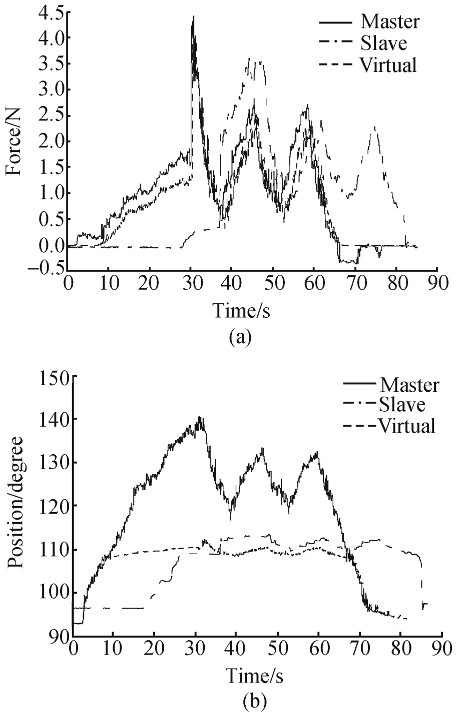

Fig. 5 Experiment result with 15 s time delay
(a) Force; (b) position

Acknowledgements This work was supported by the High-Tech Research and Development Program of China (No. 2002AA742048) and National Natural Science Foundation of China (Grant No. 60475034 and 60643007).

References

1. Anderson R J, Spong M W. Bilateral control of teleoperators with time delay. IEEE Transaction on Automatic Control, 1989, 34(5): 494−501
2. Antal B R, Kim S W. The phantom robot: Predictive display for teleoperation with time delay. IEEE International Conference on Robotics and Automation, Cincinnati: IEEE Robotics and Automation Society, 1990, 1: 546−551
3. Kotoku T. A predictive display with force feedback and its application to remote manipulation system with transmission time delay. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. Raleigh: IEEE Industrial Electronics Society, 1992, 1: 239−246

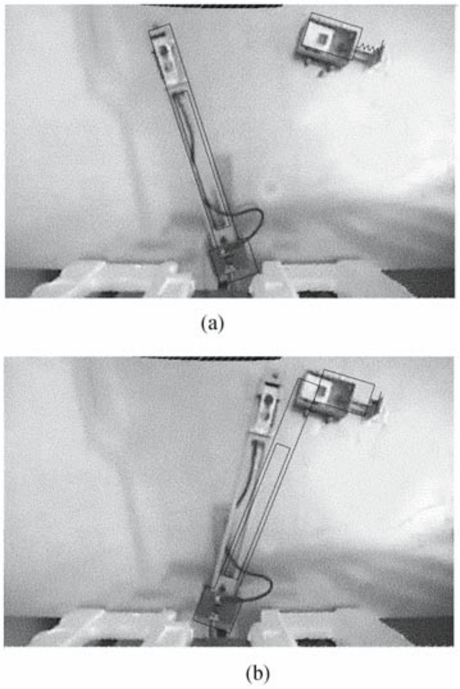

Fig. 6 Geometrical changes of environment model
(a) Before the contact; (b) contact

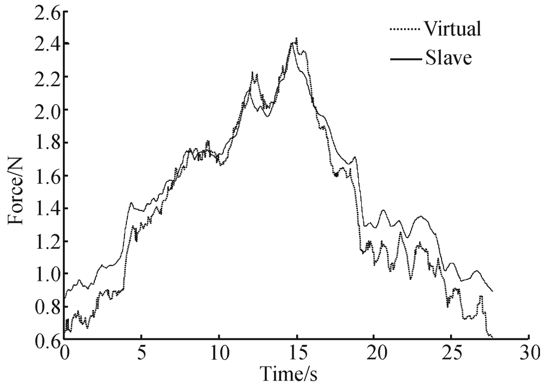

Fig. 7 Error analysis

4. Rosenberg L. Using of virtual fixture to enhance telemanipulation with time delay. In: Proceedings of ASME Winter Annual Meeting, New Orleans: ASME, 1993, 49: 29−36
5. Tsumaki Y, Uchiyama M. A model-based space teleoperation system with robustness against modeling errors. In: Proceedings of IEEE International Conference on Robotics and Automation. Albuquerque: IEEE Robotics and Automation Society, 1997, 2: 1594−1599
6. Burdea G C. The synergy between virtual reality and robotics. IEEE Transaction on Robotics and Automation, 1999, 15(3): 400−410
7. Erickson D, Weber M. Contact stiffness and damping estimation for robotic systems. International Journal of Robotic Research, 2003, 22(1): 41−57
8. Diolaiti N, Melchiorri C, Stramigioliz S. Contact impedance estimation for robotic systems. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. Sendai: IEEE Robotics and Automation Society, 2004, 3: 2538−2543
9. Love L J, Book W J. Force reflecting teleoperation with adaptive impedance control. IEEE Transactions on Systems, 2004, 34(1): 159−165