Available online at www.sciencedirect.com
ScienceDirect
Chinese Journal of Aeronautics 22(2009) 325-333
Chinese Journal of Aeronautics
www.elsevier.com/locate/cja

Virtual Reality-based Teleoperation with Robustness Against Modeling Errors

Jiang Zainan*, Liu Hong, Wang Jie, Huang Jianbin

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China

Received 28 May 2008; accepted 3 September 2008

Abstract

This article investigates virtual reality (VR)-based teleoperation with robustness against modeling errors. VR technology is an effective way to overcome the large time delay during space robot teleoperation. However, it depends highly on the accuracy of model. Model errors between the virtual and real environment exist inevitably. The existing way to deal with the problem is by means of either model matching or robot compliance control. As distinct from the existing methods, this article tries to combine model matching and robot compliance control. On one hand, the status of the virtual robot is corrected by using the position sensor data from robot joints before and during teleoperation, and the pose of the virtually manipulated object is obtained with visual recognition technology. On the other hand, compliance control algorithms of impedance control based on joint torque sensors and hybrid position/force control based on a wrist sensor have been executed in order to eliminate the small sustaining model errors. A VR-based teleoperation system of satellite on-orbit self-serving is built up. In order to verify the proposed method, an experiment deploying the solar panel troubled by malfunction is carried out through teleoperation. It shows that the large model errors are removed with the model matching method and the adopted compliance control is robust against the remaining small model errors.

Keywords: space robot; teleoperation; virtual reality; model error; visual recognition; compliance control

1. Introduction

It is well known that during space robot teleoperation, the operator's performance is seriously degraded due to large time delay. The time delay includes signal transmission delay caused by limits on the speed of light (radio transmission) and data processing delay at sending and receiving stations and satellite relay stations. For earth orbit teleoperation, the round trip time delay (i.e. the time between sending a discrete signal and receiving any feedback related to the signal) approaches 6 s[1].

The time delay in the force feedback teleoperation causes instability[2]. For the short time delay, R. J. Anderson, et al.[3] introduced the scattering theory of the two-port network to bilateral teleoperation, and proposed a passive control to maintain stability and transparency of the teleoperation system with time delay. G. Niemeyer, et al.[4], by means of the concept of wave variable, adopted the energy flow theory and maintained the stable system at the arbitrary delay. However, the stability and transparency of the delayed teleoperation system are mutually exclusive, so that, especially, at large time delay (5-7 s) in space teleoperation, ensuring stability would lead to serious decrease of transparency. Therefore, bilateral control is generally not usable in teleoperation with larger time delay.

Recently, teleoperation based on virtual reality (VR) has been developed to tackle the problem of large time delay[5-7]. A virtual environment model is created in the computer of ground site, and predictive display is provided to the operator from virtual model. Unfortunately, an exact match between the virtual world and the real world can never be guaranteed, which would make perfect performance of teleoperation risky.

To attain the consistency of the virtual and real worlds, augmented reality (AR) is used to approach model matching with error calibration techniques[8-10]. Using AR, the computer-generated predictive three-dimensional (3D) graphics are superimposed onto the real images coming from the remote environment; then the virtual model can be calibrated and the error can be

*Corresponding author. Tel.: +86-451-86412330.
E-mail address: flyjzn_2006@yahoo.com.cn
Foundation items: National Natural Science Foundation of China (60675054); National High-Tech Research and Development Program (2006AA04Z228); "111" Project (B07018)

doi: 10.1016/S1000-9361(08)60106-5

eliminated partly. However, it is of utmost difficulty to design or calibrate a virtual world with precise resemblance to the real world.

On the other hand, control schemes robust against model errors have attracted broad attention from researchers. Y. Tsumaki, et al.[11-12] proposed a new approach to VR-based teleoperation, which tolerated the geometrical and dynamic errors between the virtual world and the real world. The master arm, virtual arm, and slave arm had their own velocities and force controllers in their systems, and the control mode could be selected and changed automatically and independently. P. K. Cheng, et al.[13] introduced a local intelligent compliance controller to deal with the interaction between the robot manipulator and the environment automatically, thereby avoiding model error influences. Model errors between virtual and real worlds can be eliminated partly with either of the above two methods. It is impossible to guarantee absolute consistency between virtual and real models with the model error matching method, and control schemes robust against model errors are only effective for small model errors in the contact tasks. However, so far researches in an attempt to combine the two approaches to tackle model errors of VR-based teleoperation are very rare.

This article proposes a satellite self-serving teleoperation system, S3TS, to accomplish the space repairing tasks. The time delay (about 7 s) between the operator and the telerobot is artificially added in the local area network. Different from the existing VR-based teleoperation systems, the above two methods are combined together to eliminate the model errors between the virtual and real worlds. First, model matching is conducted using the visual and joint position sensors to calibrate the virtual model. Second, control algorithms which are robust against model errors are used to overcome the existing residual small model errors.

This article is organized as follows. Apart from Section 1, Section 2 introduces the structure of the S3TS. The virtual model matching of the robot and the manipulated object is described in Section 3. Section 4 introduces the control algorithms inclusive of proportional differential (PD) position control, impedance control based on joint torque sensors and hybrid force/position control based on a wrist force sensor. After that, Section 5 shows the experimental results of fulfilling the solar panel deployment using the proposed different control modes.

2. Structure of S3TS System

Since deployable appendages of the satellite, such as solar panels and antennas, may suffer malfunctions after launch[14], which makes the satellite fail to work normally and lose huge amounts of money, an idea of S3TS system is brought forward to settle the problem. It proposes to mount a space robot system on the satellite to remove mechanical troubles.

Fig.1 shows the structure of S3TS system based on VR technology. It consists of two sides: operator and remote robot, which are connected through a local area network (LAN). In order to simulate the large time delay during space teleoperation, the artificial delay (about 5-7 s) is incorporated into the system by data buffer technology.

At the operator side, a high-fidelity virtual model of the remote robot and environment is created for the predictive display.

As a comfortable, effective, high-precision 6 degree of freedom (DOF) optical controller, the 3D connection space ball 5000 is connected to the computer by USB for robot manipulation. The cyber glove, a fully instrumented glove that provides up to 22 high-accuracy joint-angle measurements with force feedback, is used to control the robot hand through the transmission control protocol/internet protocol (TCP/IP) communication. The video-displayed computer displays

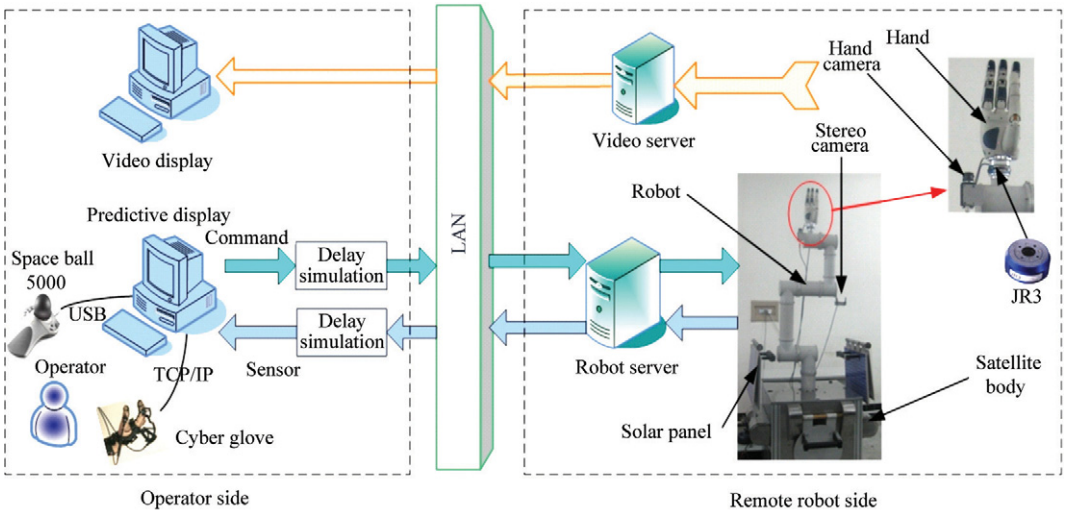


Fig.1 System of S3TS structure.

the pictures of the hand and stereo cameras. The modules of delay simulation placed both in the command and sensor channels are set to be approximately 3.5 s for simulating the 7 s round trip delay. In teleoperation, with the help of video display, the operator wears the glove with force feedback to manipulate the virtual hand and uses the space ball to move the virtual robot. The input commands are sent to the remote executive real robot system if no errors happen in the virtual environment. With the predictive display, the operator can avoid the "move and wait" strategy and markedly shorten the time to complete the task.

The remote robot side consists of a 4 DOF arm and a dexterous hand with 13 DOF[15]. The stereo cameras on the third joint of the robot are used for monitoring, whereas the local camera on the end joint is used for recognizing the manipulated object. As a six-axis force/torque sensor produced by JR3 Inc., the JR3 sensor is installed at the wrist of the robot to measure the contact force between the hand and environment. The visual server is used for visual computation, video compression, and transmission, whereas the robot server is used for controlling the real robot system and communicating with the operator side.

3. Model Matching of Virtual Environment

It is known to all that match between the virtual mode and the real world is crucial for VR-based teleoperation. However, to build a precise model of the real world is almost impossible. Thus, model matching in the S3TS system is meant to reduce large model errors between the virtual environment and real environment through the position sensor data from the robot joints and the visual sensor.

3.1. Virtual environment modeling

It is the basic work of S3TS to build up a virtual model of the real environment in the computer by using VR technology, for high-fidelity modeling can effectively improve the operator's performance of predictive display. The virtual environment modeling consists of geometrical modeling and kinematical modeling. The former is to build up the 3D graphic model of the robot and environment whereas the latter is to define the kinematic parameters based on the robotics.

Used for geometrical modeling of the virtual environment, the Open Inventor 6.0, developed by Mercury Computer Systems Inc., providing the power and functionality of OpenGL at an object-oriented level, is an object-oriented, cross-platform 3D graphics toolkit for the development of interactive 3D graphics applications using C++, .NET, or Java. Its user-friendly application programming interface (API), its extensible architecture, and its large set of advanced components—all provide developers with a high-level platform for rapid prototyping and development of 3D graphics applications.

However, it is impossible to build up a complex environment model (especially for modeling complex 3D surface) to directly use the API of the Open Inventor. The 3D modeling software, ProE, is very convenient for building up complex 3D models. Therefore, the proposed system takes advantage of both ProE and Open Inventor for virtual environment modeling. The virtual environment of S3TS consists of the robot system, the hand, and the satellite body, as shown in Fig.2.

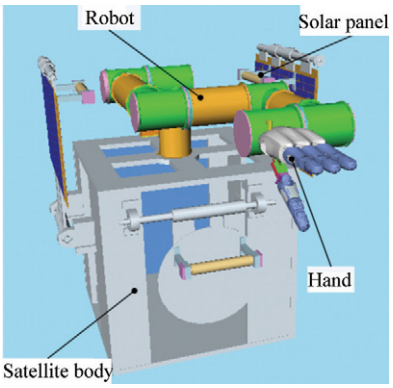


Fig.2 Virtual model of S3TS.

As the virtual environment functions only to provide the operator with the appearance of real world, it is not necessary to include the complex structure of inside parts, without which the size of the whole virtual model would be decreased from 110 MB to 35 MB allowing the model to load much faster for the VR-based teleoperation program.

3.2. Model matching of virtual robot

In the S3TS, the virtual robot is matched using the position sensor data from the robot joints. The matching is carried out before and during teleoperation.

First, it is very important to make the situation of the virtual model the same as that of the remote world before teleoperation. After having communicated with the remote real robot system, the joint position data from the remote real robot and hand can be obtained and fed back to the operator side, and then the status of the virtual robot and hand can be set to be the same as the real robot, which is necessary before the correct teleoperation and later the telemanipulation can be conducted.

Second, during the process of teleoperation, there might be some errors caused by command data missing and/or defective control methods (e.g. impedance control fails to obtain high-precision position data). To reduce these errors, it is necessary to clear up the difference between the virtual and remote robots.

In the S3TS, a rebuilt virtual model of the feedback robot is established to display the remote environment to the operator for telemanipulation. Then it is very intuitionistic for the operator to know the difference between the virtual and real robots through the 3D

graphical models.

The error between the virtual and remote robots is defined as

$$e = \|\boldsymbol{X}_{\mathrm{v}} - \boldsymbol{X}_{\mathrm{c}}\| \tag{1}$$

where $\boldsymbol{X}_{\mathrm{v}}$, $\boldsymbol{X}_{\mathrm{c}}$ are the position and orientation vectors of the virtual and real robots, respectively.

When the error e is big enough, i.e., $e \geq E$, where E is the threshold for updating, a warning message generates to remind the operator to update the situation of the virtual robot to the real one, which could be executed in a manual or automatic manner.

3.3. Model matching of virtual manipulated object

In most cases of teleoperation, the position and pose of a manipulated object in the real world is not fixed. Then the virtual manipulated object should be set to be the same as the real one. However, as there are usually no sensors arranged on the manipulated object, the information about it can not be acquired except through a visual sensor.

In the S3TS, a camera is mounted at the end joint of the robot for recognizing the position and pose of the real manipulated object.

The problem of illumination in space is complex and full of variables. Light is weak when the satellite is in the shadow of the earth but very strong when it faces the sun. However, the power of a computer disposed in the satellite is limited. Therefore, the method to recognize a target object should be taken into account.

The target object is equipped with an optical marker composed of two rhombuses, as shown in Fig.3.

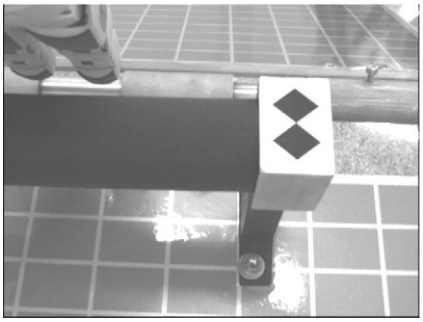


Fig.3 Target object.

First, the image caught by the camera is converted into a binary image by threshold segmentation. Then the contour is extracted by a simple morphological operator (see Fig.4). A chain code is used to represent the extracted contour.

Then, polygonal approximation is used to represent the shape. Polygons can be identified as the target object if they meet the following two conditions: ① The polygon is a parallelogram; ② Two corresponding vertices of two polygons are very close with a certain distance, and the areas of the two polygons are almost the same.

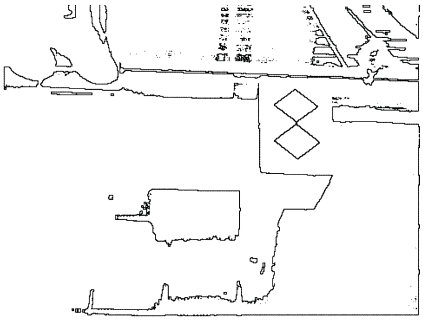


Fig.4 Extracting of image contour.

As the cooperative object is found, the vertices of rhombus are the control points. Fig.5 shows that the optical markers are recognized. Then the position and pose of the target object can be obtained through calculation of the marker.

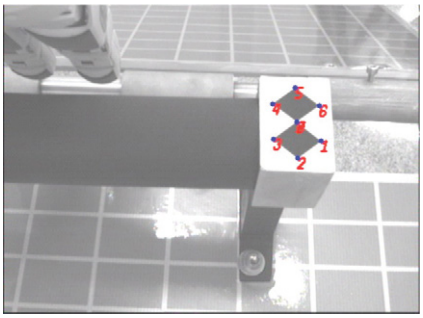


Fig.5 Recognized target object.

The position and pose of the manipulated object in world coordinates can be computed through transformation

$$^{\mathrm{O}}T_{\mathrm{w}} = {}^{\mathrm{c}}T_{\mathrm{w}}\ {}^{\mathrm{O}}T_{\mathrm{c}} \tag{2}$$

where $^{\mathrm{c}}T_{\mathrm{w}}$ is the position and pose of the camera in the world coordinates, which can be calculated by the forward kinematics of the robot, and $^{\mathrm{O}}T_{\mathrm{c}}$ the position and pose of the manipulated object in the camera coordinates, which has been obtained by using the visual recognition algorithm.

4. Control Algorithms of Remote Robot

The PD position control, the Cartesian impedance control based on joint torque sensors, and the hybrid position/force control based on a wrist force sensor have been implemented in the remote robot. This section will discuss these control algorithms in detail.

4.1. PD position control

As a simple tried and tested basic control algorithm, a PD position control has been arranged in the S3TS robot. The control $\boldsymbol{u}$ is computed by

$$\boldsymbol{u} = k_{\mathrm{pp}}(\boldsymbol{q} - \boldsymbol{q}_{\mathrm{d}}) + k_{\mathrm{pd}}\dot{\boldsymbol{q}} \tag{3}$$

where k_{pp} is the proportional gain, k_{pd} the differential gain, $\boldsymbol{q}$ the actual joint angle, and $\boldsymbol{q}_d$ the desired joint angle.

4.2. Cartesian impedance control based on joint torque sensors

As a compliance control scheme, the Cartesian impedance control does not control the desired position and force directly, but ensures the force and position to meet the desired dynamic relationship by regulating the end impedance of the robot.

As there is a torque sensor in each joint of the robot, the Cartesian impedance control is responsible for achieving the compliance of the robot. Fig.6 illustrates the structure of Cartesian impedance controller.

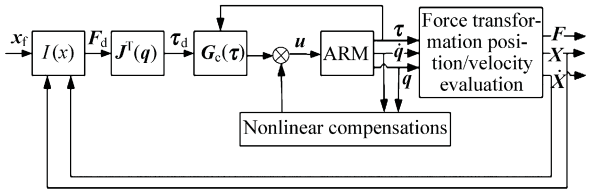


Fig.6 Structure of Cartesian impedance controller.

The aim of the Cartesian impedance controller is to establish a mass-damper-spring relationship, expressed by Eq.(4), between the Cartesian position $\Delta\boldsymbol{x}$ and the Cartesian force $\boldsymbol{F}_d$.

$$\boldsymbol{F}_d = \boldsymbol{M}\Delta\ddot{\boldsymbol{x}} + \boldsymbol{D}\Delta\dot{\boldsymbol{x}} + \boldsymbol{K}\Delta\boldsymbol{x} \tag{4}$$

$$\Delta\boldsymbol{x} = \boldsymbol{x}_f - \boldsymbol{x} \tag{5}$$

where $\boldsymbol{M}$, $\boldsymbol{D}$, $\boldsymbol{K}$ are positive definite matrices representing the virtual inertia, damping and stiffness of the system respectively, $\boldsymbol{x}$ and $\boldsymbol{x}_f$ the actual and desired Cartesian position vectors.

The Cartesian force $\boldsymbol{F}_d$ from Eq.(4) is transformed into desired joint torques $\boldsymbol{\tau}_d$ directly through Eq.(6).

$$\boldsymbol{\tau}_d = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{F}_d \tag{6}$$

where $\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})$ is the transposed Jacobian.

The joint torque controller $\boldsymbol{G}_c(\boldsymbol{\tau})$ is a PD force controller with the gains $k_{ip} = 10$ and $k_{id} = 0.1$.

The nonlinear compensations include gravity compensation $\boldsymbol{g}(\boldsymbol{q})$ and friction compensation $\boldsymbol{f}_{fric}$.

Then the control can be acquired through Eq.(7) with joint torque controller and the nonlinear compensations.

$$\boldsymbol{u} = \boldsymbol{\tau}_d - k_{ip}(\boldsymbol{\tau} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{\tau}_d) - k_{id}\dot{\boldsymbol{\tau}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{f}_{fric} \tag{7}$$

4.3. Hybird force/position control based on wrist force sensor

In the S3TS, a 6 DOF force/torque sensor (JR3) is mounted on the wrist of the robot to measure the contact force between the hand and the environment. Then, a hybrid force/position controller is developed based on the JR3 sensor, which works in parallel with the robot impedance controller by fulfilling the position correction, and allows the contact force to be controlled in a desired manner.

(1) Hybrid force/position control algorithm

Fig.7 shows the model of the solar panel. In deploying the solar panel, the operator uses a space ball to control the robot, actually, to control the point of C (X_c, Y_c) to follow the arc trajectory. Then the central angle of the handler can be calculated through

$$\theta = \arctan(X_c / Y_c) \tag{8}$$

In order to make the solar panel deploy effectively at the position C, the robot should be made to move in the D_p direction, not in the D_f direction. Therefore, in the hybrid force/position control, D_p is defined as the position direction which the control should ensure, and D_f the force direction which the control should guarantee.

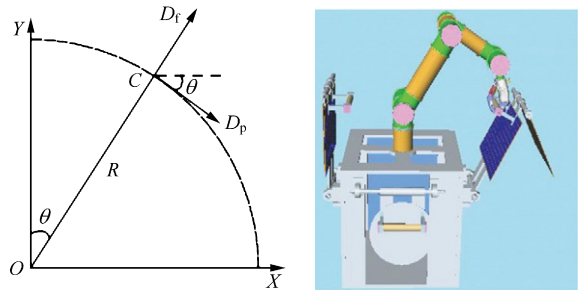


Fig.7 Model of solar panel.

From Fig.7, D_p and D_f can be found below.

$$D_p = -\theta \tag{9}$$

$$D_f = \pi/2 - \theta \tag{10}$$

Fig.8 shows a block diagram of the hybrid force/position control based on JR3 contacting its environment.

In Fig.8, $g(s)$ and $h(s)$ represent the transfer functions of the force controller and the robot impedance controller system, respectively. The contact force is measured and fed back in an external loop encircling the impedance control loop.

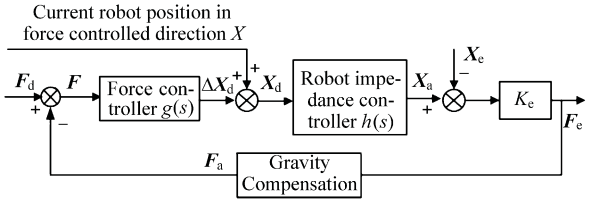


Fig.8 Force control based on JR3.

The contact stiffness K_e is the total stiffness of robot and environment measured at the contact point. The position correction based on the contact force is computed by

$$\Delta\boldsymbol{X}_d = \boldsymbol{F} / K_e \tag{11}$$

(2) Gravity compensation

In the hybrid force/position control algorithm, $\boldsymbol{F}_a$ should be the contact force between the robot and the environment. However, there is a large extra force caused by disturbance from the hand mounted on the end of the wrist sensor. In order to measure the true force in contact with the environment, this additional force of the hand should be removed at first.

The additional force of the hand includes the hand gravity and inertial force due to unsteady movement of the robot.

Later in the experiment, the robot is controlled at a constant speed so as to obviate the inertial force caused by unsteady movement to the most extent.

Obviously, the additional force would change with the pose of the hand. However, experiments have evidenced its effects so small as to be omitted if the hand does not move too fast.

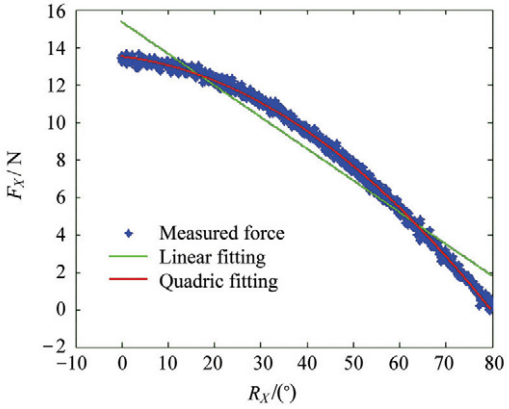
As such, the additional force mainly stems from the gravity of the hand, which is found to be only related to the robot pose. Online gravity compensation is used to remove the additional force.

First, the robot is controlled to move at a constant speed with different poses (from 0° to 80°). The force measured by JR3 is recorded synchronously. Fig.9 shows the linear least squares fitting results of force data in X and Y directions, respectively. It is clear that the quadric fitting is more appropriate. Then the dependence of force $\boldsymbol{F}$ upon the poses of the robot R_X can be described by

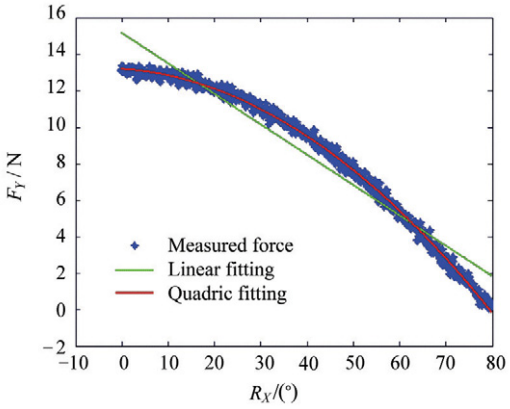
$$\|\boldsymbol{F}\| = AR_X^2 + BR_X + C \tag{12}$$

where A, B, C are the coefficients of the curve fitting, respectively.

Second, when the robot is controlled through teleoperation, the real contact force is the online measured force from which the additional force has been removed using Eq.(12). The compensation results of the online measured force data in X and Y directions are shown in Fig.10. Fig.10 shows that the mean of the compensated values is about 0 N, and the largest value is around 2 N. And the compensated force is adequate for force control after the mean filter.

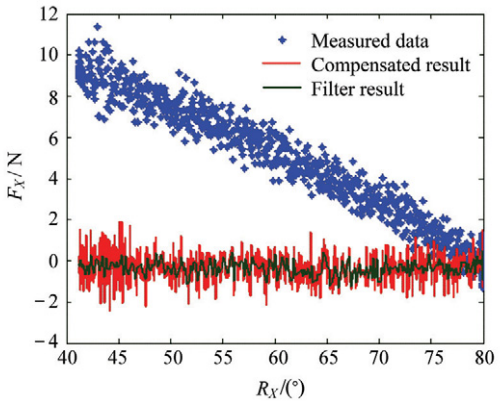


(a) X direction

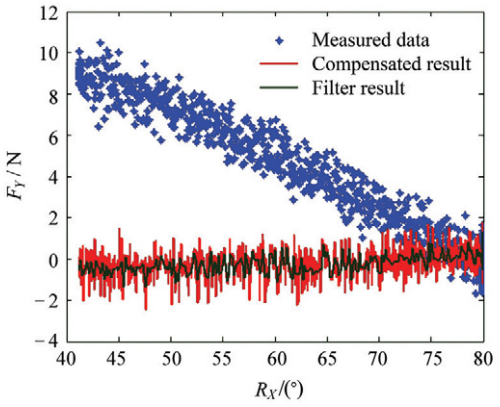


(b) Y direction

Fig.9 Curve fitting of force data.



(a) X direction



(b) Y direction

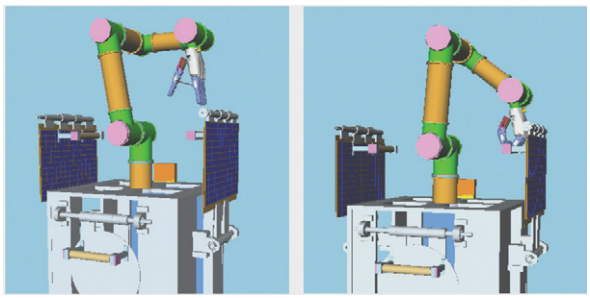
Fig.10 Gravity compensation results.

5. Experimental

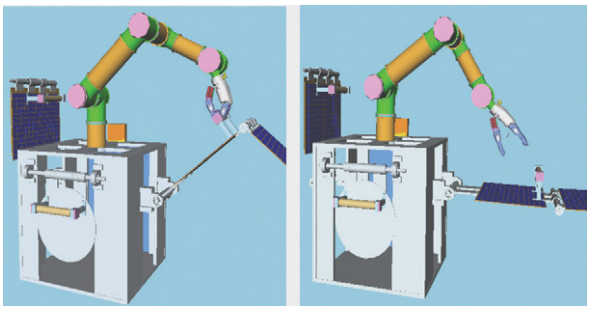
An experiment has been carried out at the lab to deploy the folded solar panel of the satellite through the LAN. The whole job can be divided into approaching the handler, grasping the handler, deploying the solar panel, and departing from the handler. Fig.11 shows the divided stages of the predictive display at the operator side. Fig.12 shows the corresponding situations of the real robot at the remote side.

In the experiment, the three different control algo-

rithms described in Section 4 have been tested and the results are shown below.

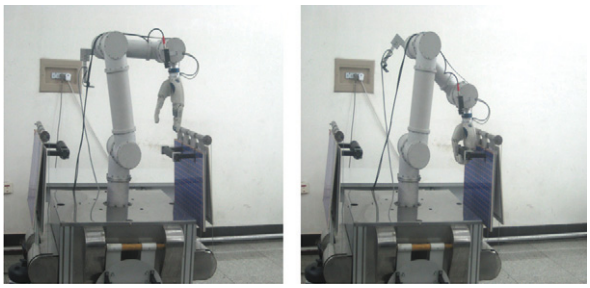


(a) Approaching (b) Grasping

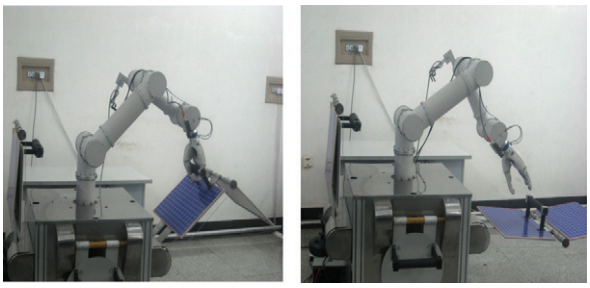


(c) Deploying (d) Departing

Fig.11 Predictive display of deploying solar panel.



(a) Approaching (b) Grasping



(c) Deploying (d) Departing

Fig.12 Remote robot of deploying solar panel.

5.1. PD position control

In the teleoperation experiment, the simple PD position control is responsible for the robot, and Fig.13 shows the contact force. The largest contact force in the force direction amounts to nearly 76 N—a dangerous one that might ruin the robot system or the solar panel of the satellite in manipulation.

In the PD position control, the robot acts exactly upon the position command from the remote teleoperator. Vulnerable to model errors, the position control is unable to eliminate even very small model errors. If the virtual model is so precise that no model error exists between the virtual and real models, the PD position control is undoubtedly appropriate for use in the VR-based teleoperation. However, this is not the case.

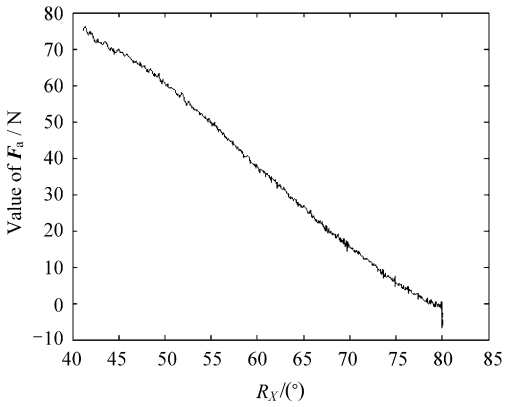


Fig.13 Contact force of PD position control.

5.2. Impedance control based on joint torque sensors

In the impedance control based on joint torque sensors, the robot receives the position command from the teleoperator, and the relationship between the Cartesian position and the Cartesian force is established.

Fig.14 shows the contact force in the force direction of impedance control. The largest force drops to about 15 N much smaller than that of PD position control (76 N). Fig.15 shows the position errors in command and real robot in X and Y directions. Different from the PD position control, the position errors are not zero because the impedance control has the ability to regulate based on the joint torque sensors in executing the telecommand. As such this algorithm has robustness against tiny model errors. However, the method has a shortcoming that it cannot obtain the accurate position and force control.

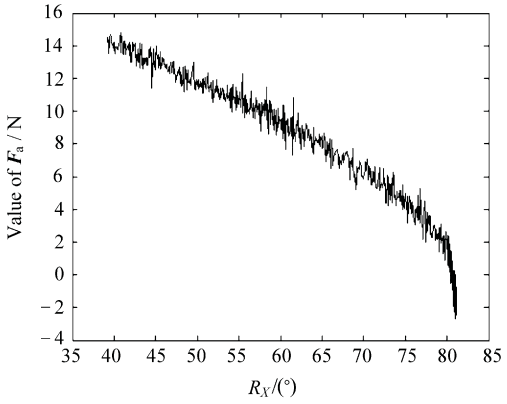
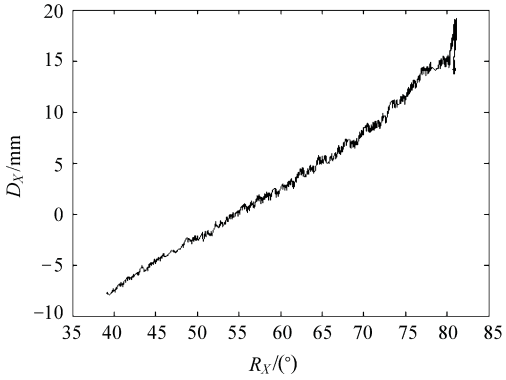
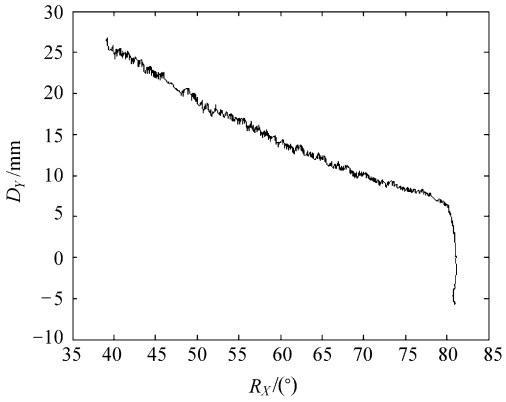


Fig.14 Contact force of impedance control.



(a) *X* direction



(b) *Y* direction

Fig.15 Errors of impedance control.

5.3. Hybrid force/position control based on JR3

In the hybrid force/position control, the robot executes not only the position command from the teleoperator, but also the corrected position command from the local force control loop at the robot side. The local force control loop calculates the position correction and allows the contact force to be controlled in a desired manner. Finally, the robot executes the sum of the operator's input and the local regulation position command.

Fig.16 shows the contact force in the force direction of hybrid force/position control. The largest contact force reaches about 9 N which is smaller than the largest force of impedance control (15 N). Fig.17 shows the regulative position generated from the local force loop. From Fig.16 and Fig.17, the position correction increases with the increase of contact force. A bandpass filter is used in the position correction.

Fig.18 shows the position errors of the command and the real robot in *X* and *Y* directions. It could be seen that this algorithm has the ability of eliminating minor model errors.

Similar to impedance control, this control algorithm is able to eliminate small model errors. Moreover, with the position correction, this control scheme shows better performance, for the least contact force in the force direction is smaller.

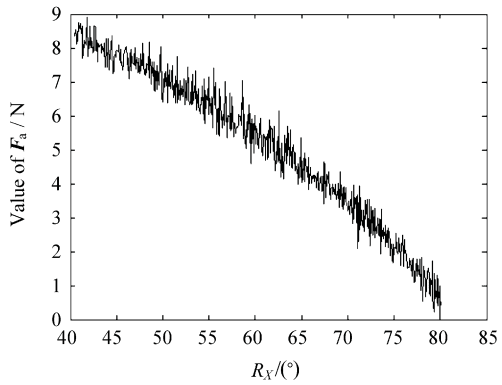


Fig.16 Contact force of hybrid force/position control.

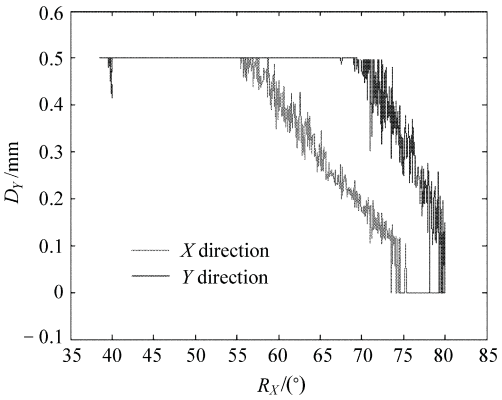
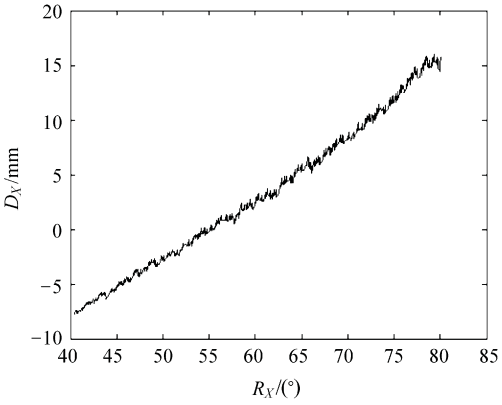
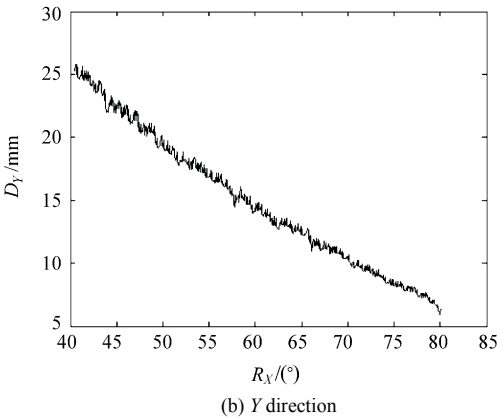


Fig.17 Command correction based on JR3.



(a) *X* direction



(b) *Y* direction

Fig.18 Errors of hybrid force/position control.

6. Conclusions

This article has proposed a method for VR-based teleoperation with robustness against model errors. In order to remove the unavoidable model errors between the virtual and real environment, a method that combines model matching and compliance control is put forward. Firstly, the virtual robot is corrected using the robot joint position sensor, and the manipulated object is recognized by computer vision. Secondly, compliance control methods of impedance control based on joint torque sensor and hybrid force/position control based on JR3 are used to eliminate the small remaining errors.

The experiment of deploying a solar panel shows that with the model matching method, large errors between the virtual and real environment could be removed, and with the compliance control, the remaining small model errors could be eliminated.

References

[1] Sheridan T B. Space teleoperation through time delay: review and prognosis. IEEE Trans Robot Autom 1993; 9(5): 592-606.
[2] Ferrel W R. Remote manipulation with transmission delay. NASA TN D-2665, 1965; 24-32.
[3] Anderson R J, Spong M W. Bilateral control of teleoperators with time delay. IEEE Trans on Automatic Control 1989; 34(5): 494-501.
[4] Niemeyer G, Slotine J J. Telemanipulation with time delays. International Journal of Robotics Research 2004; 23(9): 873-890.
[5] Chantanakajornfung P, Gil H. VR simulator for nano SMMS teleoperation over the delayed networks. SICE-ICASE International Joint Conference. 2006; 4826-4831.
[6] Rodriguez N, Jessel J P. A virtual reality tool for teleoperation research. Virtual Reality 2002; 6(2): 57-62.
[7] Yang X L, Chen Q. Virtual reality tools for internet-based robotic teleoperation. Proceedings of the 8th IEEE International Symposium on Distributed Simulation and Real-Time Applications. 2004; 236-239.
[8] Xiong Y J, Li S Q. Predictive display and interaction of telerobots based on augmented reality. Robotica 2006; 24(4): 447-453.
[9] Marin R, Sanz P J. A predictive interface based on virtual and augmented reality for task specification in a Web telerobotic system. IEEE International Conference on Intelligent Robots and Systems. 2002; 3005-3010.
[10] Kim W S, Schenker P S. Advanced operator interface design with preview-predictive displays for ground-controlled space telerobotic servicing. Proceedings of the International Society for Optical Engineering. 1993; 96-107.
[11] Tsumaki Y, Hoshi Y. Virtual reality based teleoperation which tolerates geometrical modeling errors. IEEE International Conference on Intelligent Robots and Systems. 1996; 1023-1030.
[12] Tsumaki Y, Uchiyama M. Model-based space teleoperation system with robustness against modeling errors. Proceedings of IEEE International Conference on Robotics and Automation. 1997; 1594-1599.
[13] Cheng P K, Young K Y. VR-based teleoperation for robot compliance control. Journal of Intelligent and Robotic Systems: Theory and Applications 2001; 30(4): 377-398.
[14] Zhang Z M. The malfunctions manual of spacecraft. Beijing: Press of Space Navigation, 1994. [in Chinese]
[15] Liu H, Meusela P. The modular multisensory DLR-HIT-Hand. Mechanism and Machine Theory 2007; 42(5): 612-625.

Biography:

Jiang Zainan Born in 1982, he is a Ph.D. candidate in mechanic engineering at Harbin Institute of Technology (HIT), Harbin, China. He received his B.S. degree in 2003 and M.S. degree in 2005 from HIT. His main research interests include space robot teleoperation, virtual reality, human-machine interaction, and mobile robot navigation.
E-mail: flyjzn_2006@yahoo.com.cn