Design and Implementation of a Telerobotic System with Large Time Delay

Wu Xiaotao*, Zhuang Jun*, Sun Zengqi**, Zhang Zhenmin+

*Room 223, Blvd. 1, Tsinghua University
Beijing, 100084, P.R.China
email: wxt062371@hotmail.com
Tel: 86-10-62781479

**Dept. of Computer Science, Tsinghua University
Beijing, 100084, P.R.China
email: szq-dcs@mail.tsinghua.edu.cn
Tel: 86-10-62785118

+Office of Expert Committee for Space Technology,
The High Technology Research & Development Program in China,
Beijing, 100034, P.R.China

ABSTRACT

This paper introduces a telerobotic system. In this system, a delay-compensating 3-D stereo-graphic simulator is implemented in SGI ONYX/4 RE^2 with SPACEMOUSE, HMD devices, Sirius Video. Programs written in Dual Robot System Simulation Language (DRSSL) can be used to control the simulating robot in graphical environment. The command sequences are generated at the same time with the movement of the simulating robot and are sent to the real robot after the simulating time delay. The images gotten from the camera are sent back to make overlapping to the simulating robot. Virtual reality technology and shared control are supported in this system. Some basic tasks are accomplished by controlling PUMA560 robot.

1. INTRODUCTION

For humans to expand permanently into space, teleoperated robots can function indefinitely in space environment. Teleoperation allows humans to participate intimately yet safely in taming the space environment. Technologies for ground-remote telerobotics have been developed to support ground-based control of space-based robots. Communication time delay between the local site operator control station, where the operator resides, and the remote site robot control system is a significant factor for ground control telerobotics. A round-trip delay of several seconds or more is expected. The time delay precludes control modes which require closed loop control between the local and remote sites. The operator can not operate the remote site robot in time according to the video image. An alternative method is to generate command sequences through interaction in a local graphical environment. By simulating the real remote site robot and environment, we can control remote robot easily and efficiently. In Section 2, a high quality graphical environment will be shown and the relating parts for controlling the simulating robot are introduced.

To verify the control effect, the graphical environment is connected to the PUMA560 robot. Two cameras are placed behind the real robot and on top of the robot to take the images and send them back to make overlapping with the graphical robot. The sensor information of each joint is also sent back to be compared to the value of the simulating robot. This part will be described in Section 3.

In Section 4, a robot real-time motion planning method based on configuration space (C-space) is proposed.

In Section 5, a Dual Robot System Simulation Language (DRSSL) is introduced, two robots can be controlled parallelly by this language.

The Section 6 will describe how to integrate all the parts into a whole system. Shared control and some basic tasks will be discussed in this section too.

This system was used to control PUMA560 robot to perform some basic tasks, this will be introduced in Section 7.

2. GRAPHICAL ENVIRONMENT

A high quality graphical environment can let operator operate the simulating robot easily and efficiently. In our system, a graphical environment was implemented in SGI ONYX with 4 CPU and 2 Reality Engine board. In the environment, the model can be displayed with texture and can also be displayed in frame or solid mode without texture. In the texture mode, the rate of the display can be about 60 frames/sec.

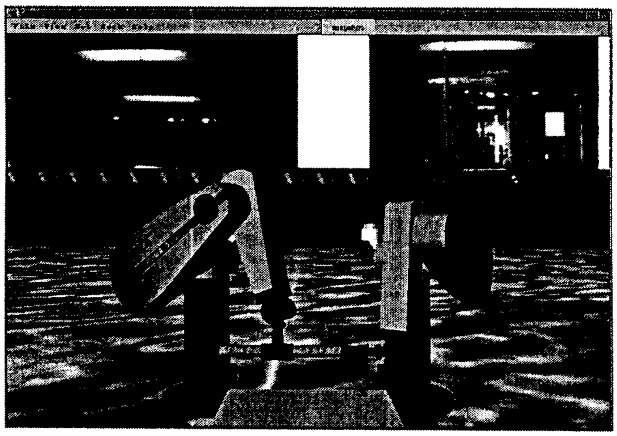


Fig. 1 Graphical Environment with Texture

There are several view-ports to let the robot can be watched in different view-point simultaneous, especially, the view-point can wonder in the environment freely and conveniently, so that the robot can be operated more accurately. i.e., When a pole wants to be plugged into a hole, it can be seen from front, top and left side simultaneously, and zoom in and zoom out are used to see the detail to make sure that the pole can be inserted into the hole. Also, Virtual Reality technology is supported. WTK is used to deal with the VR programming. Three-dimensional stereo glasses are used to get 3-D stereo-graph.

The SPACE MOUSE is used to control the simulating robot in the environment. SPACE MOUSE has six-degree-of-freedom (dof) so that joint-to-joint control can be used to operate PUMA560 (with six-dof), for a robot with seven or more degree-of-freedom, we can control the position and orientation of the grasp in Cartesian Space with the shared control introduced.

3. TO VERIFY THE CONTROL EFFECT

The PUMA560 robot was connected to the graphical environment. The graphical robot and the real robot have the same structure, so the joint-to-joint control is used to control the real robot. When the simulating robot move, the command was sent to the real robot after the simulating time delay to control the real robot to move as the simulating robot.

There are two methods to verify the effect of the control. One is to compare the feedback sensor information to the information of the simulating robot. The other is to do overlapping between the image sent back by the camera and the graph of the simulating robot.

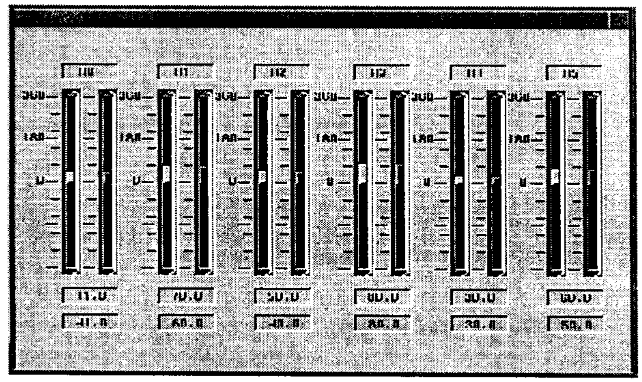


Fig. 2 Sensor Information Visualization

For the first method, an interface (Fig. 2) to make sensor information visualized was implemented to let operator to get the result directly by the sense.

Fig. 3 Overlapping between image and graph

For the second method, to do overlapping between the image and the graph, we should first calibrate the camera and demarcate the view point and then to overlap the graphical robot on the image. In the Fig. 3, the background is the image gotten from the camera in real time. The frame drawn by line is the graph of the simulating robot created by the computer. If the frame can overlap the image of the real robot all the time, we can say the control is successful.

4. REAL-TIME MOTION PLANNING

Because a space robot usually operates in uncertain environment, the motion planning should be performed in real-time. A planning method based on configuration space (C-space) is introduced into this system. [13]

5. DUAL ROBOT SYSTEM SIMULATION LANGUAGE (DRSSL)

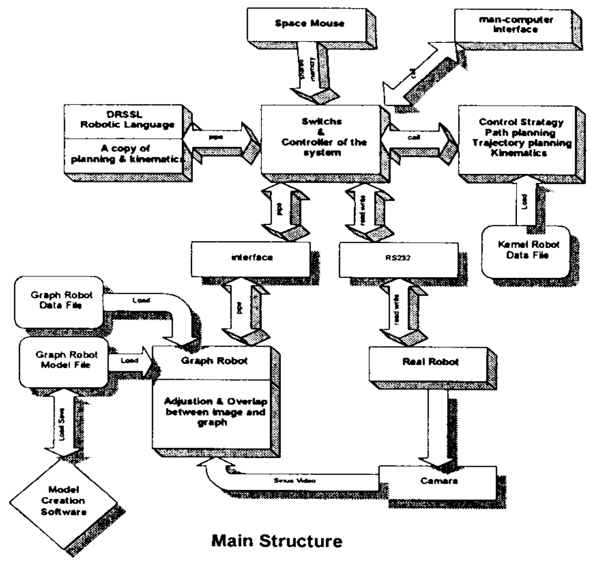
Compared to VAR-II language, Dual Robot System Simulation Language (DRSSL) has additional capacity to support simulation of dual robot system. The programs written in DRSSL can be compiled to execution files and can also be interpreted to execute. Both on line and off line programming are supported by DRSSL. It also includes command for SPACE MOUSE control. The Object Oriented Technology is used to implement the language, and the users can develop their DRSSL programs in Object Oriented Architecture, so the reusability and the expandability of the program are enhanced greatly.

There are two parts of DRSSL. One is an integrated environment. This environment includes compiler, executor and debugger. The compiler translate the robot language to the executable code, a series of atomic operation, which can be understood by the executor. The debugger can support setting breakpoint, running step by step, viewing the value of the variable and changing the value of the variable dynamically. The other part of DRSSL is an interpreter, the command can be execute immediately through this part.

6. SYSTEM INTEGRATION

The whole system is designed by Object Oriented Design (OOD) method. It has clear structure and can be updated easily. The structure is as Fig.4.

In the structure, the coordinator is very important. Every part communicates with the coordinator instead of communicating to each other. By this mean, the shared control is easily to be supported. And the interface is clear.



Main Structure

Telerobotic System with Long Time Delay

Fig. 4 The Structure of the System

In the procedure of designing, we consider the real robot and the graph robot as the equal object, in their class, the same interface was defined. By this mean, the data flow is clearly to be understood.

7. THE BASIC TASKS

The system was used to perform the following tasks:

1. plug a pole into a hole
2. assemble a structure from several identical cube parts
3. two arms grasp a object together
4. two arms to do plugging operation coordinately

In the tasks, the real robot could repeat the graphical robot's movement after the simulating time delay. The frame of the graphical robot can cover the image of the real robot all the time. The result is successful.

8. CONCLUSION

Many aspects of robotics were integrated into the system. Key technologies of the system have been:

1. High quality graphical simulation, the model was displayed with texture.
2. Dual Robot System Simulation Language
3. Real-time robot motion planning, the cost of the planning is less than 1 second.

4. By adjusting and overlapping of the camera images and the computer graphs, the effect of the control can be judged.

To improve the performance of the system, many technologies and methods will also be used into this system such as distributed calculation, etc.

REFERENCES

[1] G.Hirzinger, K.Landzettel, Ch.Fagerer, "Telerobotics with Large Time Delays - the ROTEX Experience", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp571-575

[2] C. Weisbin, D. Perillard, "Jet Propulsion Laboratory Robotic Facilities and Associated Research", Robotica(1991) volume 9, pp 7-21

[3] Michael C. Nechyba, Yangsheng Xu, "SM^2 for New Space Station Structure: Autonomous Locomotion and Teleoperation Control", 1994, IEEE, pp1765-1770

[4] M.B.Leahy, Jr.S.B.Petroski, "Unified Telerobotic Architecture Project Program Overview", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp240-244

[5] P.Michelman, P.Allen, "Shared Autonomy in a Robot Hand Teleoperation System", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp253-259

[6] T.Tarn, A.Bejczy,C.Guo, N.Xi, "Intelligent Planning and Control for Telerobotic Operations", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp389-396

[7] J.Perret, C.Proust, R.Alami, R.Chatila, "How to Tele-program a Remote Intelligent Robot", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp397-404

[8] M.R.Stein, C.P.Sayers, R.P.Paul, "The Recovery From Task Execution Errors During Time-Delayed Teleoperation", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp547-554

[9] A.K.Bejczy, P.Fiorini, W.S.Kim, P.Schenker, "Toward Integrated Operator Interface for Advanced Teleoperation under Time Delay", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp563-570

[10] S.Lee, K.Jeong, "Design of Robust Time-Delayed Teleoperator Control System", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp1413-1420

[11] M.Wanner, T.Herkommer, "Off-line Programming for the Aircraft Cleanning Robot "SKYWASH", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp1972-1979

[12] Tsumaki, Y., Uchiyama, M., "Predictive display of virtual beam for space teleoperation", Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 96. Robotic Intelligence Interacting with Dynamic Worlds, vol.3 pp.1544-1549

[13] Li Wei, Ma Chenyu, Chen Zushun, Cao Qi and TSO. S. K., "Sensor-based Obstacle Modeling in Configuration Space for Manipulator Motion Planning", 1996 IEEE Int. Conf. of Multisensor Fusion and Integration for Intelligent Systems, Washington.