Published in
Proceedings of the 5th International Conference on Manufacturing Technology
Beijing (1st to 3rd November 1999)

VR TELEROBOT SYSTEM

Tan Hock Soon
Lee Keng Hong
Kwong Kwok Kuen
School of Engineering
Temasek Polytechnic
Singapore

Abstract

Technological breakthroughs have allowed automation to replace many jobs that are repetitive in nature. However, there are many tasks that are non-repetitive, unpredictable and hazardous to human health. These tasks have to be performed by a remote manipulator, otherwise known as a telerobot. Telerobotics refers to the use of robots "at a distance".

With manufacturing going global, there is an increased need for robots to be deployed over several countries. To service and diagnose these robotic equipment involve a lengthy process of sending experts to the application sites to diagnose, order spare parts and fix the problem. If this process could be shortened, downtime of the machine and charges incurred would be saved. By virtue of its operation, the telerobot is able to provide information about itself to the remote operator.

Virtual Reality (VR) is the science of integrating man with information. It consists of three-dimensional, interactive, computer generated environments. Hence, conceptualisation of complex or abstract systems is made easier by providing a "natural" man-machine interface.

This paper attempts to present an application that allows a robot to be controlled remotely through a VR interface. By doing so, it is also able to tap information about the robot operating conditions. The following is discussed in the paper:

- Description of the Telerobot system
- Approach to Implementation
- Integration Techniques/Data Exchange
- Results

1. Introduction

Robots are mechanical arms that perform physical actions on their environment; they are characterized by flexibility (versatility), as a given robot can execute a variety of tasks or execute a given task in a variety of ways. Robots can be "intelligent" or self-adapting, so they can react to changes in the environment and take corrective actions in order to perform their tasks successfully [3]. Robots that have these attributes are usually used in manufacturing. However, robots are also used in unstructured environments such as remote maintenance or in areas hazardous to human health. This means that the robot has to be controlled via teleoperation. One of the difficulties associated with teleoperation is that the human operator is remote from the robot; therefore the feedback data may be insufficient for correct control decisions. Hence, a telerobot is described as "a form of teleoperation in which a human operator acts as a supervisor, intermittently communicating to a computer information about goals, constraints, plans, contingencies, assumptions, suggestions and orders relative to a limited task, getting back information about accomplishments, difficulties, concerns, and as requested, raw sensory data – while the subordinate robot executes the task based on information received from the human operator plus its own artificial sensing and intelligence" [7].

To operate a telerobot, new technology has to emerge to take advantage of complex new robotic capabilities while making systems more user-friendly.

Virtual Reality (VR) can be described as the science of integrating man with information. It consists of three-dimensional, interactive, computer generated environments [21]. The purpose being to represent information through synthetic experience. A virtual model differs from that produced by a Computer Aided Drafting (CAD) system in two basic ways. Firstly, the user can interact with the objects in the model, secondly, behaviours can be associated with objects in the model.

An intuitive interface between man and machine is one which requires little training (particularly in complex operational procedures) and offers a working style commonly used in human interaction with environments and objects. In other words, the human interacts with elements of his task by looking, holding, manipulating, listening, speaking, moving, using as many of his natural skills as are appropriate, or can reasonably be expected to be applied to a task [7].

Robotics is a very important field especially for the engineering, manufacturing and the scientific communities; however VR applications in these areas are still in the early stages of development. This paper attempts to present a system that allows a robot to be controlled remotely through a VR interface.

2. Review of Research - VR Techniques for Telecontrol

Research shows that robot telecontrol is based on either the Augmented Reality (AR) or the Telepresence techniques.

Augmented Reality is described by Azuma [2] as a variation of VR which allows the user to see virtual objects superimposed upon or composited with the real world. Drascic [6] describes in his paper the ARGOS (Augmented Reality through Graphic Overlays on Stereo-video) system that allows an operator to indicate a precise destination for the telerobot, or to indicate a path for it to follow by using a virtual pointer in a three dimensional video space. Krishnan and Gopalsamy [9] have also used the AR technique by superimposing two dimensional graphical control elements (buttons, switches, graphs, trends, etc.) onto a live video scene. Cannon [4] used what he called a virtual tool, which appears as graphic representations of robot end-effectors interwoven into live video. The operator direct tasks involving attributes in the same natural way that supervisors direct human subordinates, for example using commands such as "put that there", "cut there", and others.

Telepresence is described in Earnshaw,Gigante & Jones [7], as "the ideal of sensing sufficient information about the teleoperator and task environment, and communicating this to the human operator in a sufficiently natural way, that the operator feels physically present at the remote site". Lim, Ang & Wong [10] made use of this technique in the control of a semi-anthropomorphic robot. The system consists of using an exoskeleton fitted to a human arm which is synchronised with the robot. Feedback is through a HMD (Head Mounted Device) which allows the user to view the work area through a HEM (Head Eye Module) mounted on top of the robotic arm. It allows the human operator to remotely control the robotic arm as an extension of the human arm. Churbuck [5] describes how a company helped to build a special telepresence spraying device. Rather than flying the limited pool of expert operators around, the device allows the experts to be brought on site via telepresence. This taps the expertise of the expert operators while maximising the use of the spraying systems around the world

Hine et.al. [8] describes the remote operation of complex robotic mechanisms in unstructured natural environments using what he called VEVI (Virtual Environment Vehicle Interface) which draws heavily on Virtual Environments with VR as the user interface. The paper describes the design philosophy and implementation of VEVI in the field of science exploration missions such as space, deep sea and volcano. The VR user interface is used to replace the "normal" method of using telemetry data to determine the state of the controlled mechanism and uplinking commands to the mechanism to achieve a desired new state. The telemetry data are often displayed as numeric streams or graphs and the uplinked commands are entered by the operators through typed text strings and/or button presses. The interpretation of the system state from the telemetry data is usually so complex that each subsystem has one or more dedicated human controllers monitoring it at all times. By using a VR interface, a relatively unsophisticated operator could comprehend the current and past state of the system quickly, to plan and review high-level commands to the system and to send commands for the system to execute. Tan & Lee [16] have shown a step by step approach in building a tele-operated robot utilising a VR user interface using commercially available software and hardware.

In all the examples cited, regardless of whether the system uses a Telepresence, Augmented Reality or Virtual Reality technique, the following conditions were observed:

a. Time Delay: When the time delay grows much larger than the dynamic time scale of the remote mechanism, the telepresence method is inefficient because of the delay between the sensing and the actuation. This delay makes the control system unstable [8].
b. Communications Bandwidth: A related problem to that of time delay is the communications bandwidth from the human controller to the remote mechanism. When a close loop control is difficult or impossible, a move-and-wait strategy or local decision making capability is incorporated into the system. The operator interface should be usable and efficient when operating over low bandwidth communication channels.
c. Efficient Command Cycles: If a teleoperator must send a dozen commands to the remote system in order to position it, the teleoperator interface is very inefficient; hence, the

Published in
Proceedings of the 5th International Conference on Manufacturing Technology
Beijing (1st to 3rd November 1999)

teleoperator must have access to high and low level commands to get maximum flexibility in sending efficient command sequences.

d. Situational Awareness: The operator interface should lead to greater situational awareness than can be accomplished with a traditional control approach.

e. Modular Components: Modern large software and hardware systems can only be efficiently maintained and extended if the modular components are written in a modular fashion. This allows the system to be designed more easily, and facilitates future modification, re-use and ease of support.

f. Use of Commercial Products: Wherever possible, leveraging from commercially available tools and products is a cost effective approach to design and build the system.

These conditions were taken into account in the construction of the VR telerobot.

3. Design

All robots have a four-level command and control structure, as shown in Figure 1. The lowest level is common to all automatic systems (robots or dedicated machines) and consists of position servoing based on internal sensor data. Here the task is to follow a given trajectory (geometrical

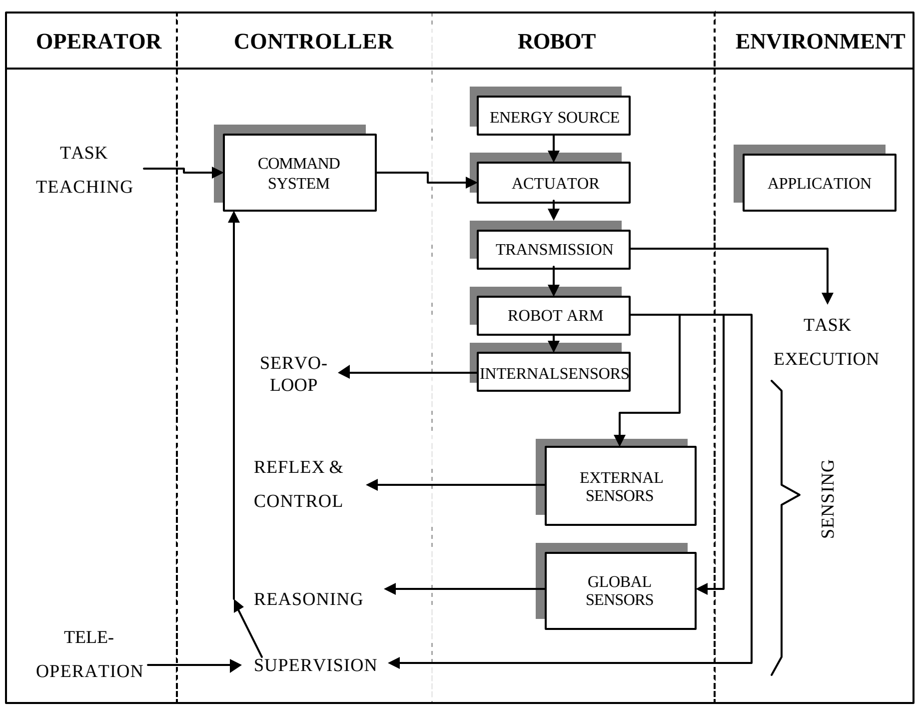

Figure 1: Robot Control (Adapted from Burdea & Coiffet [3])

or in the force domain), based on inputs from the higher control levels. The second control level, the "reflex" level, helps the robot avoid obstacles or converge towards its goal. The third level of control is the "reasoning" loop. It uses global sensors (eg. Vision, ultrasound, etc.) to analyze the state of the robot environment and extract an optimal trajectory to the goal. Finally, the highest level of control is the human operator. He both "teaches" the robot new tasks, and subsequently intervene to rescue the robot should problems arise.

The operation of the telerobot is merely an extension of level 4 using a new user interface. The design architecture of the telerobot is shown in Figure 2. The design is based on the findings in section 2, as well as the approach suggested in Tan & Lee [16]. The approach is as follows:

a) Get Robot Details: This includes information on the dimensions of the robot, the limits of movement, the behavior and properties. In addition, information regarding interfaces, both hardware and software in the robot controller that could allow it to exchange data with the external environment is essential
b) Determine Interface Technique & Methods of Data Exchange: Once the technique of tapping data from and sending data to the robot controller is determined, it is then possible to decide the methods to be used in exchanging data with the robot controller. It is usual that a program be written in the controller whose ultimate function is to update the robotic variables (e.g. joint positions, gripper state, etc.) constantly. This information is in turn accessed over the network. The method of accessing the robotic variables as well as the method of enabling the VR model to access these variables have to be determined. This may mean choosing a VR software that will allow the model to send and receive data using the same protocol. Under the Windows platform, common protocols used for such data exchanges include: DDE (Dynamic Data Exchange), OLE (Object Linking and Embedding) and ActiveX.
c) Choose a Programming Language to build the interface: The language chosen must suit the purpose of the robotic application and should be able to use the data exchange protocol to communicate with the robot controller program as well as the VR model.
d) Build VR Model & Verify Model Behaviour: Having chosen the VR software that will allow exchange of data into and out of the model, the next step is to build and verify against the physical system.
e) Build VR-Channel Interface: Having constructed the VR model, codes are then inserted in the model to enable handshaking and data transfer to and from the interface program(known as the channel). Codes are also inserted codes into the channel to do a similar function. It is not necessary at this stage to implement all the data variables, merely a few; to prototype the interface.
f) Test the VR-Channel Interface: The interface is tested until it is error free.
g) Build Channel-Robot Interface: A program is written in the robot controller to enable this exchange of data. Codes are inserted into the channel program to do the same. Again, it is not necessary to implement all the data variables.
h) Test the Channel-Robot Interface: The interface is tested again until it is error free.
i) Build Interfaces with Full Capability: Once the methodology of data transfer is stabilized, tapping of all necessary data is implemeted.
j) Lastly, the full integration is tested.

Published in
Proceedings of the 5th International Conference on Manufacturing Technology
Beijing (1st to 3rd November 1999)

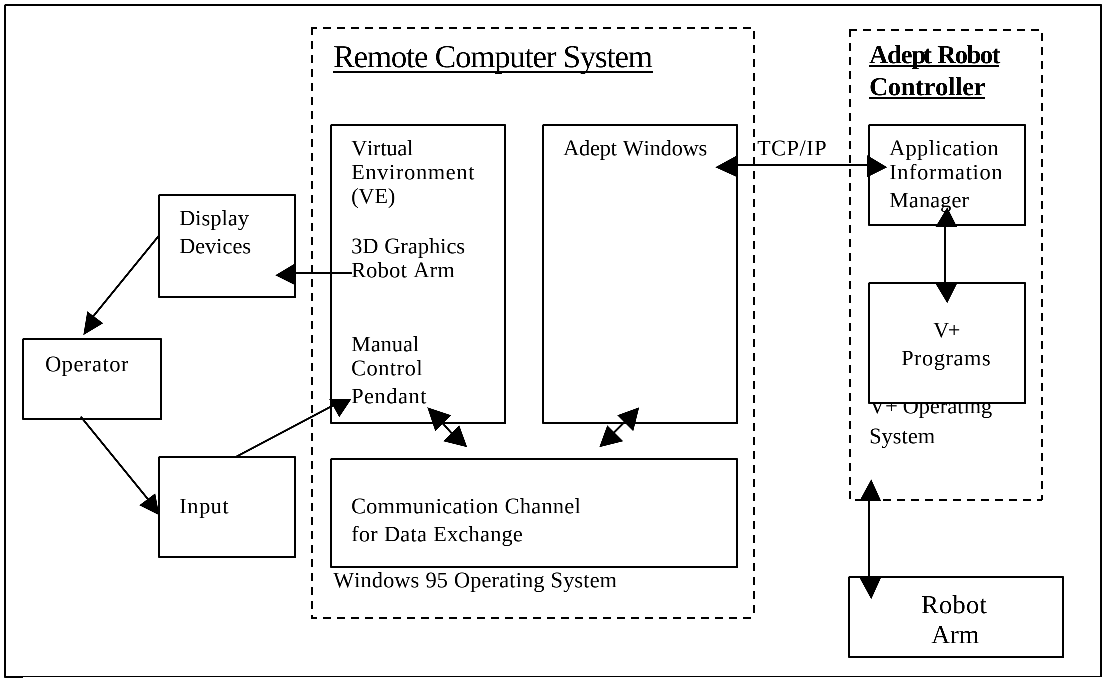

Figure 2: System Design

In this project, the robotic arm used was an Adept 550 SCARA Robot. The controller runs on the V+ Operating System and comes with the V+ programming language. The robot controller also has the option to run AIM (Application Information Manager), a graphical development environment based on a database. The database allows its data to be shared through DDE (Dynamic Data Exchange). DDE is a data exchange format commonly used in automation equipment to share data with Window applications.

As shown in Figure 2, the V+ programs can have access to various system parameters in the robot and also update the AIM database. Through TCP/IP, the data in the database is then shared with the Virtual Environment (VE) via an interface program called the communication channel. This works both ways: hence, control parameters can be sent from the VE to the AIM database via the communication channel and TCP/IP. Another set of V+ programs then extract these parameters from the database and update the robot. Therefore, only the command and status data is sent through the network resulting in small time delays and bandwidth requirements. This approach however, can only be used for a structured environment. In unstructured environments, other forms of feedback have to be incorporated. One such form is through visual feedback from cameras. Naturally, if fully coloured video feedback is needed, the bandwidth requirement would have to increase accordingly.

Published in
Proceedings of the 5th International Conference on Manufacturing Technology
Beijing (1st to 3rd November 1999)

4. System Architecture

Figure 3 shows the physical setup of the Telerobot system. The robot controller is connected to a network hub. A user is able to access data from the remote robot controller first through its IP address and then via Adept Windows and the DDE protocol in the AIM database as mentioned in the previous section. Using a similar method, the user can also send commands to the robot.

The four cameras are used to monitor the front, two sides and plan views of the robot. Following the concept of keeping the control distributed, these images are captured and processed on another computer and the images are then sent across the network to the operator. The purpose of the camera is to verify and act as a safety feature to ensure that unforseen events can be handled safely. The rationale is also to insulate the data transmission which runs at an unpredictable rate (high or low) unlike the camera feedback which runs at a constant rate.

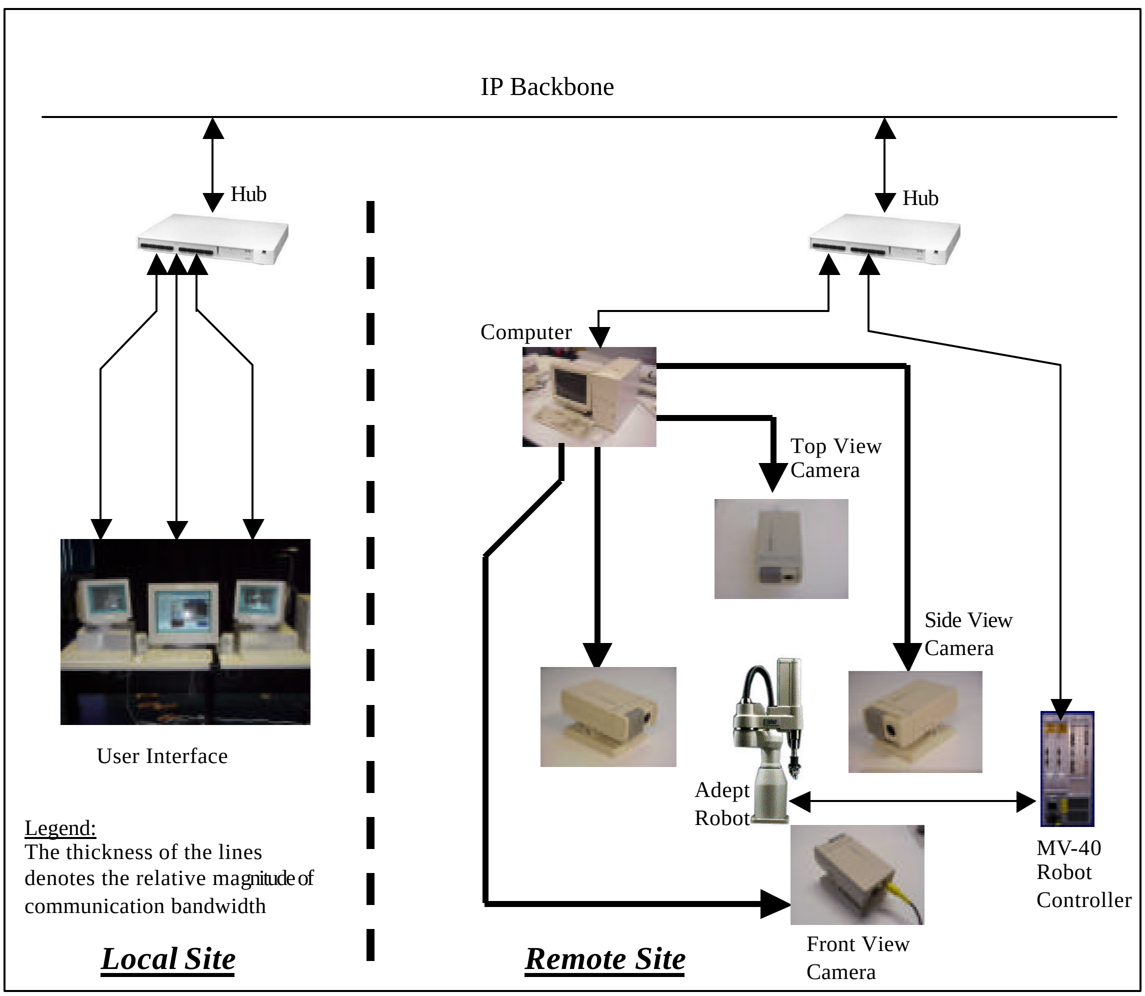

Figure 3: Schematic Diagram of System

Published in
Proceedings of the 5th International Conference on Manufacturing Technology
Beijing (1st to 3rd November 1999)

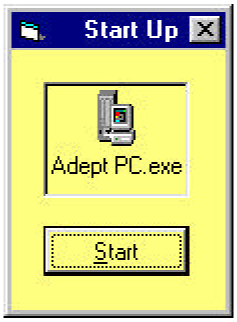

Figure 4: Startup

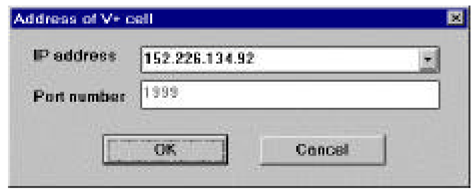

Figure 5: Connection to Robot Controller

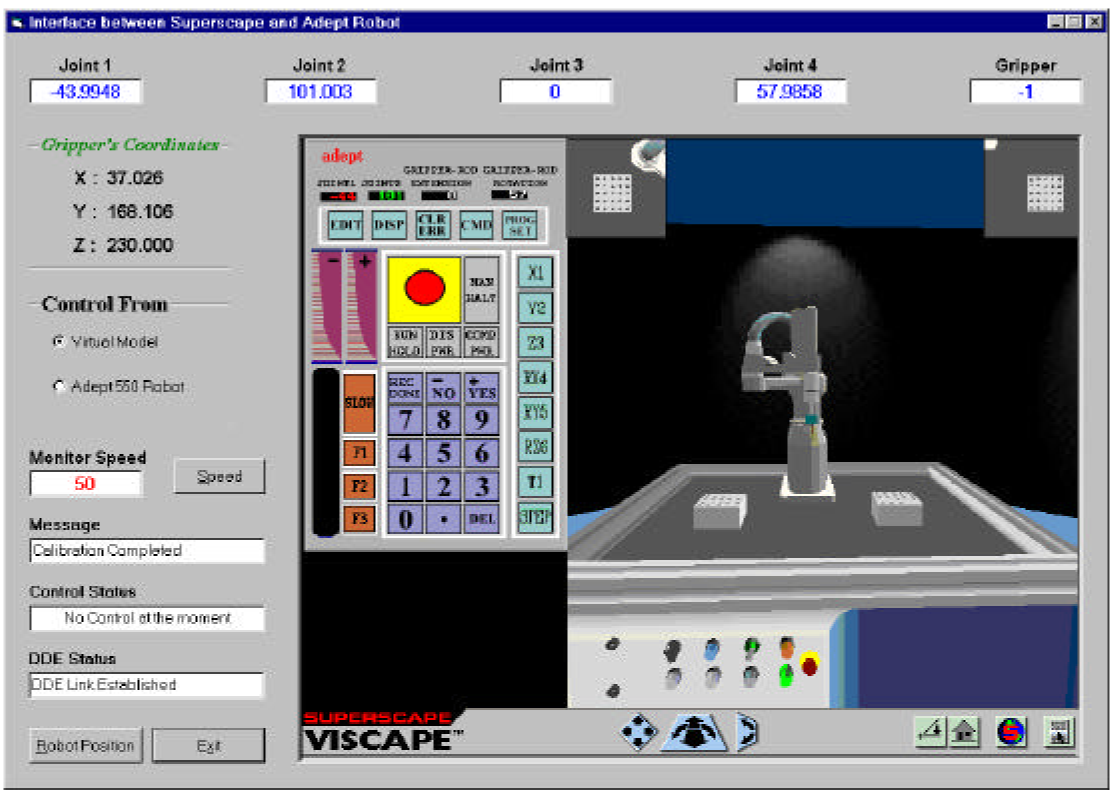

Figure 6: VR-Based User Interface

Figure 4 and 5 show the user interface during startup and its connection to the robot controller via IP address. The operator user interface is shown in Figure 6. VR is used to convey the robot's status as well as act as an interface for issuing commands to the robot. In VR, the operator is mainly interacting with the 3D renderer. This process displays the robot as an interactive 3D graphical display to the user. The 3D renderer is not a simulator, but is rather a system display and represents the actual state of the remote robot.

The 3D renderer is basically an interactive graphical simulation, which displays the state of the robot. During the system design phase, several options were considered in implementing the 3D renderer. The approach taken was to use commercially available software as much as possible. This led to the decision to use Superscape VRT 5.5 because it is a stable product which is easy to use and has the required APIs to integrate with the robot. It is also low in cost, works on the PC platform and satisfies the graphical requirement in this project.

5. Data Exchange

Data exchange starts immediately once the user switches on the "power switch" located on the panel in front of the robot in the VE. This immediately establishes DDE links between the VE in Superscape, the channel created in Visual Basic forms and the AIM database as shown in Figure 7. This event synchronizes the local VE environment and the remote robot. Next, the data channel from the VE to the remote robot controller is established.

As soon as an event occurs on the arm, a V+ program (get.values) running on the robot controller updates the AIM database. This in turn triggers the DDE protocol and that bit of data is immediately used to update the VE over the network. The reverse works the same way, that is, when a command is issued in the VE, it is sent over the network and updates the AIM database. This in turn triggers a V+ program (put.values) which then executes the data.

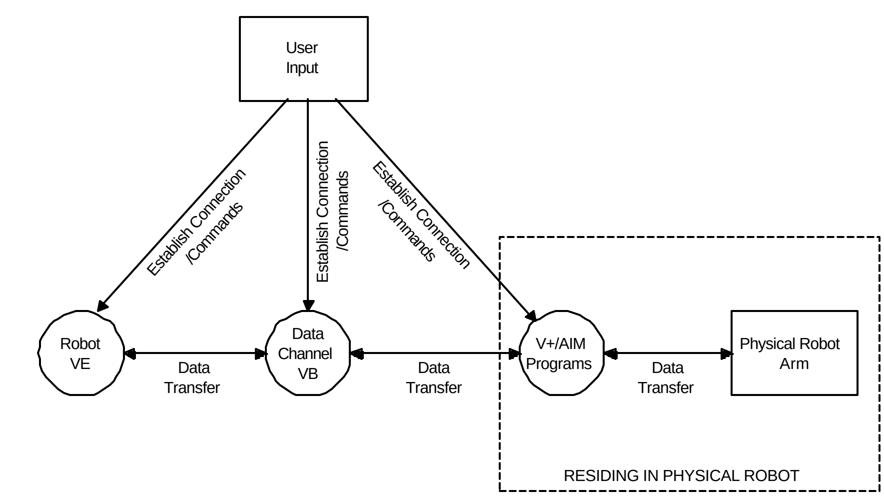

Figure 7: Data Exchange Process

A sample program for establishing communications in the VE is as follows:

Published in
Proceedings of the 5th International Conference on Manufacturing Technology
Beijing (1st to 3rd November 1999)

```
if (activate (me, 1))
{/**Open connection to Data Channel**/
  d=ddeopen ("Adept", "frmAdept");
  g=1;
  ddeunlnk (d, "txtStatus");
  /**Send message to respective text box**/
  ddesend (d, "txtStatus", "DDE Link Established", 25);
  ddesend (d, "txtMsg", "Calibration in Process", 25);
}
```

Figure 8: Sample SCL Program in VE

The VE environment detects if an event is triggered and opens a connection to the data channel. It then proceeds to update the data channel.

The data channel is the interface between the VE and the robot controller over the network. Figure 9 shows how the connection is made:

```
module = "aim\tele\"
topic = "Adept|User1"

    'Link to Joint 1 of the Adept Robot
    txtJoint1A.LinkMode = 0 'Cancel all current link
    txtJoint1A.LinkTopic = topic
    txtJoint1A.LinkItem = module + "joint1"'Track joint 1 positio
    txtJoint1A.LinkMode = 1 'Link Automatic
```

Figure 9: Establishing Data Channel Connection

The topic variable holds the name of the DDE server in the AIM database with the topic name User 1. The database item name is found in the “tele” module and the reference for joint1 is “joint1”. So whenever “joint1” in the “tele” module is updated, the data is automatically updated in the data channel. When a command is issued to move joint 1 of the robot in the VE, it is again updated in the data channel which then updates the database.

The “HERE” command in figure 10 reads the current position of all the robotic joints and places them into #loc. #loc is then decomposed and updated in the array jt[1] location. Similarly, the gripper, power status and speed of the robot are updated in their various locations in the database.

Published in
Proceedings of the 5th International Conference on Manufacturing Technology
Beijing (1st to 3rd November 1999)

```
.PROGRAM get.values(args[], error)

; read joint values
HERE #loc

; extract real individual components
DECOMPOSE jt[1] = #loc     ; joints are extracted
                             automatically one by one

; gripper status
gripper = SIG(1)

robot_power = SWITCH(POWER)
robot_speed = SPEED(1)

RETURN

.END
```

Figure 10: Data Synchronisation

6. Findings and Conclusion

Preliminary observation of the telerobot over a MAN environment using a 100MBPS network indicated that data transfer to and from the VE interface and the telerobot is in real time. More concrete testing will be conducted to measure the actual timing over the network in the next phase of the experiment as well as the network traffic during the test. The feedback mechanism using four separate video cameras over the same network has a latency problem. When the video signal was reduced from full colour to monochrome, the communication latency was observed to be about 0.5 seconds. Again, more concrete measurement needs to be performed on the video feedback. The telerobot should also be tested on a high bandwidth network to test its performance on such a network.

In summary, the project has shown it is possible to use VR as a user interface for teleoperation as well as diagnosis. It is an efficient means of allowing the operator to quickly comprehend the remote situation and act decisively. The telerobot was also built using commercially available software and hardware. Feedback via video cameras experienced a lag in the operator interface. This coincides with findings from many researchers, for example [2] [3] and [8]. A continuation of this work is in "Applied Reality" to develop control over the Internet using a standard browser. In addition, work in unstructured environments using a combined VR and Telepresence Approach is being explored.

Published in
Proceedings of the 5th International Conference on Manufacturing Technology
Beijing (1st to 3rd November 1999)

Acknowledgements:

The Management of the School of Engineering, Temasek Polytechnic for their support and vision

Singapore Totalisor Board Funding Committee for the partial funding of this project under the Proposal " Manufacturing in Cyberspace"

References:

[1] "AdeptNet User's Guide, Ethernet for the Adept MV Controller", Adept Technology, Inc., November 1996.

[2] Azuma, R.T., "A Survey of Augmented Reality", Presence: Teleoperators & Virtual Environments, Vol. 6 Issue 4, pp 355 – 385, August 1997.

[3] Burdea, G. & Coiffet, P., "Virtual Reality Technology", John Wiley & Sons Inc., 1994.

[4] Cannon, D.J. & Thomas, G., "Virtual Tools For Supervisory and Collaborative Control of Robots", Presence: Teleoperators & Virtual Environments, Vol. 6 Issue 1, pp 1 – 28, February 1997.

[5] Churbuck, D.C., "Applied Reality", Computer Graphics, Vol. 150 Issue 6, pp. 486-489, September 1992.

[6] Drascic, D., "Stereoscopic Video and Augmented Reality", Scientific Computing and Automation, 9 (7), pp. 31 – 34, June 1993.

[7] Earnshaw, R.A., Gigante, M.A. & Jones, H., "Virtual Reality Systems", Academic Press Limited, 1994.

[8] Hine, B., Hontalas, P., Fong, T., Piguet, L., Nygren, E., Kline, A., "VEVI: A Virtual Environment Teleoperations Interface for Planetary Exploration", SAE 25th International Conference on Environment System, San Diego, CA, July 1995. http://img.arc.nasa.gov/papers/ices95_hine.pdf

[9] Krishan U.A. & Gopalsamy S., " Enhancement of Industrial Control System using Digital Video Techniques", Proceedings of Virtual Reality Technology and Internet Applications (VRTIA'98), Singapore, pp. 13-1 – 13-4, July 1998.

[10] Lim S.Y., Ang M. & Wong H.Y., "Telepresence Robot Control Using Virtual Reality Instrument", Proceedings of Virtual Reality Technology and Internet Applications (VRTIA'98), Singapore, pp. 5-1 – 5-8, July 1998.

[11] Mahoney, D.P., "Better than Real", Computer Graphics World, Vol. 22, Issue 2, pp. 32 – 38, February 1999.

[12] "MotionWare User's Guide Version 3.0", Adept Technology, Inc, October 1995.

[13] Potter, B., Maxwell, T. & Scott, B., "VISUAL BASIC SUPERBIBLE" Second edition, Waite Group Press™ ,1994.

[14] "Release Notes for AIM Version 3.1", Adept Technology, Inc., January 1997.

[15] Sheridan, T.B., "Telerobotics, Automation and Human Supervisory Control", MIT Press, 1992.

[16] Tan H.S. & Lee K.H., "An Approach to Robot Telecontrol using VR", Product Innovation & Technology Management'99, Hong Kong, pp.A1-A9, 29-30 June 1999.

[17] Tan H.S. & Ward M.D., "Virtual CIM System: A Reality in CIM Training", Proceedings of Virtual Reality Technology and Internet Applications (VRTIA'98), Singapore, pp. 2-1 – 2-10, July 1998.

[18] Teh T.A. & Zheng D.M., "Virtual Training in Manufacturing", Proceedings of Virtual Reality Technology and Internet Applications (VRTIA'98), Singapore, pp. 4-1 – 4-7, July 1998.

[19] "V+ Operation System User's Guide Version 12.1", Adept Technology, Inc, September 1997.

[20] "VRT for Windows User Guide", Superscape Inc., 1998.

[21] Warwick, K., Gray, J. & Roberts, D., "Virtual Reality in Engineering", IEE, 1993.

[22] Wolfe, A., "What in the World: Robots on the Web?", Electronic Engineering Times, Issue 820, pp. 1-2, 24 October 1994.