

# An Interactive Virtual Reality Simulation System for Robot Control and Operator Training<sup>1</sup>

Nadine E. Miner  
Sharon A. Stansfield  
Sandia National Laboratories  
Albuquerque, NM

## Abstract

*Robotic systems are often very complex and difficult to operate, especially as multiple robots are integrated to accomplish difficult tasks. In addition, training the operators of these complex robotic systems is time-consuming and costly. In this paper, a virtual reality based robotic control system is presented. The virtual reality system provides a means by which operators can operate, and be trained to operate, complex robotic systems in an intuitive, cost-effective way. Operator interaction with the robotic system is at a high, task-oriented, level. Continuous state monitoring prevents illegal robot actions and provides interactive feedback to the operator and real-time training for novice users.*

## 1 Introduction

A complex, multi-robot system is being developed at Sandia National Laboratories as a part of a DOE-sponsored Environmental Restoration and Waste Management (ER&WM) project known as the Advanced Controls Manipulation Laboratory (ACML). In this laboratory, a demonstration test bed is being developed for the technologies which will be required for the retrieval of hazardous waste from Underground Storage Tanks (UST), such as those located in Hanford, WA. A Cincinnati-Milacron (CM) robot will provide the gross positioning movement, while a specially developed Schilling Titan robot arm will provide the fine motion control. Specially designed tools will be utilized to remove waste from the tanks. Due to its application-specific nature, training on, and operation of this complex robotic system will be difficult and costly. A

virtual reality (VR) based interface, where the operator is fully immersed in the graphical environment of the robotic system, may be used to address some of these problems. In this paper, a VR system being developed in conjunction with the ACML/UST project is described. Through the VR system, operators interact with simulations of the real robots in an intuitive, natural way. The VR system can be set up to simulate both normal and exceptional behaviors to enhance the training of operators. Once completed, the VR system will allow robot commands to be captured, previewed and downloaded to the actual robot system for execution

In section 2 of this paper, the motivation for utilizing a VR interface for complex robotic system control is discussed. The architecture of the VR system being developed for use with the ACML/UST project is described in section 3. Section 4 presents a discussion of the major components of the VR system being developed. Finally, section 5 contains conclusions and a discussion of possible future enhancements to the VR system.

## 2 Virtual reality for robotic control

Virtual reality is a type of human-computer interface. With VR, the user is immersed in an environmental simulation with which they can interact [Helsel 91]. As the computer-generated environmental models become more realistic, and the user interaction more intuitive, the virtual world becomes more of a reality. The VR system being developed for robot control at Sandia allows participants to interact with realistic simulations of complex robotic systems in a natural way using task-level voice commands and hand gestures. Due to this natural interaction and task-level orientation, participants may be

<sup>1</sup>This work was performed at Sandia National Laboratories supported by the US Department of Energy under contract number DE-AC04-76DP00789.

trained to use the system easily and robot commands may be generated, previewed and executed rapidly. The user is immersed in the graphical virtual environment through a stereo viewer which tracks the user's head position and orientation. As the operator's view changes, the graphics are updated so that the feeling of immersion inside the virtual environment is achieved. When users are immersed in the virtual environment, they can get any view of the graphical model, in this case a robot workcell, by simply looking at the desired destination and 'walking' or 'flying' to the location. This type of human-computer interaction is much more intuitive than a flat screen display using a mouse to change the view.

Several flat screen, graphical programming systems have been developed at Sandia which allow an operator to program robotic systems through interaction with graphical menus, a 3-D spaceball, and various other devices [McDonald 93]. Although these systems have made great progress in improving the human-robot interface, several limitations still exist: lack of accurate depth perception, lack of non-visual feedback (audio, force, etc.) and awkward control interfaces (mouse, keyboard, teach pendant and spaceball). VR has the potential to move the human-robot interface to a new, intuitive and user-friendly level [Davies 93].

In addition, VR can be used to train robot operators by taking the physical robots out of the loop. By using a VR training system, operators can be trained before the actual robot hardware is available. During training, the operators progress can be monitored and real-time feedback on their performance can be provided. Training scenarios are easily set-up, including emergency situation procedure training. By having the operator experience the emergency situation in the virtual world, they are much better prepared to carry out the correct procedures if the actual emergency should arise. Trainers can also interact with the trainees in the virtual simulation system to further enhance the training exercise.

Related work includes that of Takahashi et al. [Takahashi 92], who proposed a virtual reality interface for robotic assembly-task teaching. Tasks were executed by an operator wearing a VPL DataGlove™, and hand

gestures were recognized and translated into robot commands. Extensive telerobotics research using VR techniques has been done by NASA-Ames Research and JPL for control of remotely deployed robots [Stark 87] [Bejczy 1980]. The VR simulation system described in this paper provides a different method of complex robotic system interaction by taking more of a task-level, supervisory approach to the problem. Penn State is developing methods for "interweaving virtual reality tools with live video scenes to direct robots" [Wang 93]. This work may be directly applicable to the VR system described here.

### **3 The VR simulation system**

#### **3.1 Target application description**

The Advanced Controls Manipulation Laboratory (ACML) is being developed at Sandia to facilitate design, integration and testing of waste remediation technologies. These technologies are being developed for application to the clean-up of massive (500K-1M gallon, 80 feet diameter) Underground Storage Tanks (UST) which have exceeded their design life. Various forms of waste are stored in the USTs including solidified crystal, sludge, vertical pipes, thermocouple trees and other material. The pipes often extend to the full height of the tanks and range from 2 to 12 inches in diameter. The ACML facility consists of two robots (Cincinnati-Milacron 786 and Schilling Titan II hydraulic manipulator attached to the wrist of the CM 786), a gripper tool attached to the end effector of the Titan II, 2 inch and 4 inch hydraulic cutting tools with ultrasonic docking sensors, optical sensors and force measurement sensors. A tank mock-up consisting of a rectangular wooden box containing sand, to simulate the tank waste, pipe stands and pipes, is used to simulate the UST environment [Davies 93]. The VR simulation system being developed will provide an intuitive, easy-to-use operator interface to this complex ACML facility.

#### **3.2 VR system architecture overview**

The VR interface system uses a combination of off-the-shelf and customized hardware and software. The

operator interface consists of an accurate, 3-D graphical model of the UST test bed which is viewed through a Fakespace, Inc. stereoscopic viewer known as the Boom. A Silicon Graphics, Inc. Crimson/RE or Indigo/Extreme workstation is used to drive the graphics and simulation and provide audio feedback. Operators issue voice commands to interact with the VR simulation. The voice commands are recognized by the Dragonwriter voice recognition system developed by Dragon Systems, Inc and are communicated to the SGI host via a serial link. Audio feedback provides the user with guidance throughout operation of the system. Figure 1 shows a block diagram of the system configuration.

## 4 VR system component descriptions

### 4.1 Operator interactions

The operator interacts with the VR system through the use of an immersive stereo viewer and voice input. Audio feedback is used to continuously guide the operator and to provide command confirmation.

**4.1.1 Immersive graphical environment:** The operator enters the virtual environment through Fakespace, Inc.'s Boom device. The Boom uses mechanical-tracking to continuously monitor the location (x,y,z) and orientation (yaw, pitch, roll) of the participant's head. The SGI workstation polls the stereo viewer 20 times per second and updates the graphical display so that the user's movements are synchronized with his/her view of the

virtual environment. In this way, the participant is given the sensation of actually being in the graphical world. The graphical environment is a geometrically and kinematically accurate model of the ACML/UST robot test bed which was generated from manufacturers' specifications and physical and sensor measurements of the actual ACML test bed. The operator moves through the graphical environment while looking into the boom by either walking or using controls on the Boom to fly. This type of interaction with the graphical environment is very intuitive. Operators become accustomed to the environment very quickly and require little instruction in how to use the system.

The VR environment was developed using the software platform CimStation™ developed by SILMA, Inc. CimStation™ is a modeling and simulation system that is used to graphically display the robot motions during robot programming, previewing and monitoring. CimStation™ also provides the capability of adding custom device drivers and user developed codes using the object-oriented Sil language on which CimStation™ is based. All of the VR application specific software has been developed in Sil and is run on the SGI workstation with the exception of the voice recognition software which is run on a PC platform. Sandia has been working closely with SILMA to enhance CimStation's™ capabilities for use as a generic VR software platform. Figure 2 shows the graphical model of the UST workcell, the actual UST robotic system, and a user interacting with the VR environment through the stereo viewer.

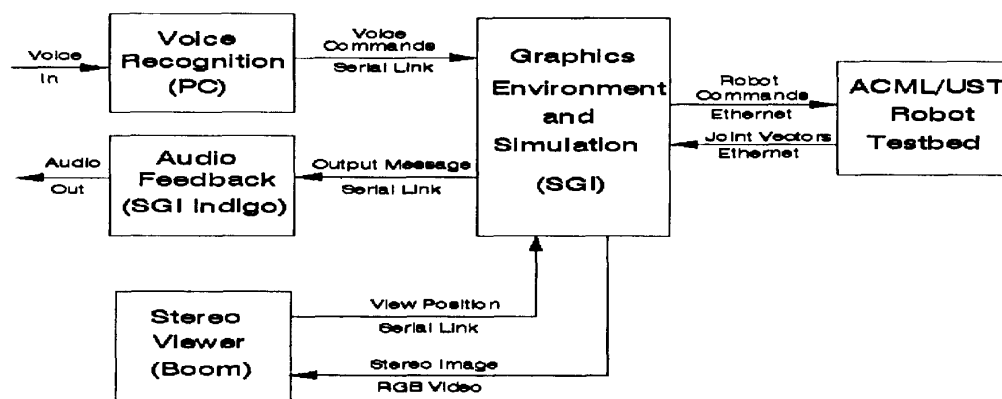
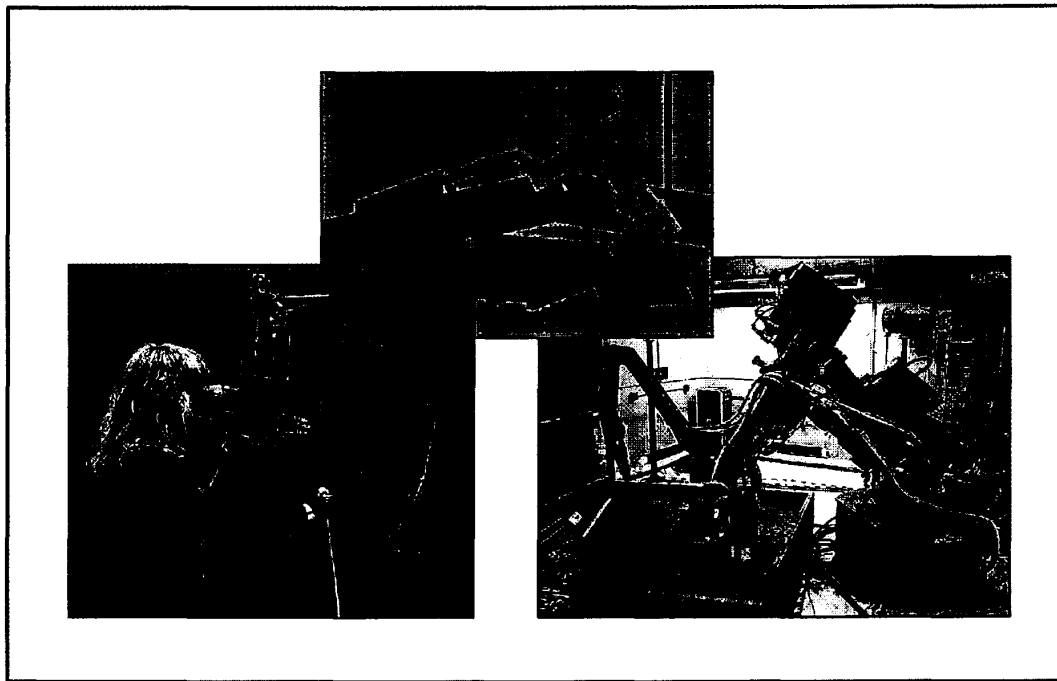


Figure 1: VR System Block Diagram



**Figure 2: User Interaction with Graphical Robot Model in VR**

**4.1.2 Voice command interaction:** The voice command input consists of a set of high-level, speaker-independent commands forming an application specific vocabulary. The vocabulary has been designed using DragonWriter™ by Dragon Systems. The DragonWriter voice recognition board operates from a PC platform. As words and phrases are recognized, ASCII characters are output by the PC to the SGI workstation through a serial link. Voice commands are processed on the SGI once per second so that voice command response is timely yet real-time graphics updates are not too severely impacted. When command sequences are obtained, the main application loop is interrupted and the voice command is verified. Once the commands are confirmed, appropriate sets of robot instructions are generated.

For the ACML/UST application there are a total of 32 words and phrases comprising the user input command set. The voice system was trained on five female and five male voices to obtain a speaker-independent voice

template; thus, new users of the application need not retrain the vocabulary. Preliminary results indicate the speaker independent voice template consistently recognizes ~85% of the voice commands, including operators with noticeable foreign accents. ASCII output is suppressed for unrecognized voice input. To minimize misinterpreted commands, the vocabulary has a hierarchical structure such that sequences of words are required to initiate command actions, such as "open" "gripper". Also, phrases are used to add distinction between commands.

The vocabulary consists of intuitive, task-level voice commands which make the system easy to learn and use. Task-level robot control has long been a goal of robotics research. Using the task-level approach, the VR application remains independent of the particular robot performing the task [Lozano-Pérez 92]. With this approach, the commands are kept generic up until the point where specific robot commands must be

communicated to the robot system for execution. From the VR perspective, various robot systems could be used to carry out the tasks simply by changing the low-level robot instruction generation routines.

An example of a task-level voice command for the ACML/UST system is, *"get cutter"*. When the command is recognized, the ASCII characters, '20', corresponding to the *"get cutter"* command are sent from the PC to the VR system on the SGI. If the command is valid in the current system state, a sequence of robot actions occur. The operator receives audio confirmation that the command was received and can observe the command execution in the virtual environment. The graphical representation of the Titan II robot arm is commanded to move the gripper tool to the cutter tool location (stored on the tool bar at a known location), the gripper tool grasps the cutter, and then the arm is moved safely back to the 'home' position. The paths for robot movement are generated using CimStation's™ internal trajectory planner. In order to execute this command on the actual robot system, the *"get cutter"* command would be downloaded to the robot control system and, at this lowest level, be translated into the set of Titan II robot and tool commands. Sensors on the physical robot system would be used for final approach and grasping operations so that inconsistencies between the simulated and physical robot paths would not be critical. The operator need not learn details of the robot kinematics nor the specific steps required to get the cutter tool; they simply need to learn the intuitive task-level voice command to initiate the operation. Other examples of task-level commands are: *"store cutter"*, *"open cutter"*, *"close gripper"*, and *"cut pipe"*.

The VR system has been designed to provide the capability of programming sequences of robot commands through a sequence of operator initiated voice commands although the integration of these commands into the VR system is still on-going. Once implemented, robot command capture will be initiated with the voice command *"begin robot programming"* and will commence with the command *"end robot programming"*. The set of task-level voice commands entered while robot programming is enabled will be accumulated for eventual

download to the ACML/UST robot system. The VR system will provide the capability of previewing the set of captured task-level commands with the voice command *"preview robot program"*. During previewing, the RIPE/RIPL command set will be simulated on the ACML/UST's robot safety system in the same manner as is done in Sandia's graphical robot programming system [McDonald 93]. The operator will view the safety simulation from the VR environment as it progresses to visually verify correct operation. Once the command set has been confirmed by the operator and the robot's safety checks, the operator will be able to download the command set to the robot workcell for execution with the voice command, *"execute robot program"*. An abort feature is planned so that the operator will be able to halt command execution if the actual and planned robot motion differ too dramatically.

**4.1.3 Audio feedback:** During operation, the operator receives continuous audio feedback from the VR system. This serves several useful purposes. First, the operator obtains confirmation of voice commands. For example, with the *"get cutter"* command, the audio system responds with the phrase *"getting cutter"* as the graphical simulation executes the command. In this way, the operator receives auditory and visual command confirmation.

The operator may also ask for *"help"* or system *"status"* at any time. The voice command *"status"* gives the status of the workcell and results in an audio sequence such as: *"The current status is: gripper closed, cutter open, cutter tool inactive"*. When the operator requests *"help"*, the current valid commands are output to the user, which might be a sequence such as: *"The current valid commands are: get cutter, open gripper, close cutter"*.

Finally, the operator can be guided through use of the system with audio feedback with varying levels of terseness. For example, the audio system lets the user know when an invalid command is received. If the operator attempts to execute a *"cut pipe"* command before the cutter tool has been obtained, the audio feedback message would be: *"You must select the cutter tool before executing the cut pipe command. Use 'get*

cutter' to select the cutter tool". If the trained user mode is selected (with the voice command: "set trained user"), the audio feedback message would be more terse: "cannot cut pipe". In the default novice user mode, the audio system provides the operator with more verbose instructions and command options. The audio feedback capability is provided by an Indigo 2 workstation via a socket server. Message pointers are sent to the audio server to initiate appropriate audio feedback messages depending on the users actions and commands.

### 4.2 System intelligence component

Several methods are possible for control and implementation of task-level tele-robotic systems. Takahashi et al. uses If-then rules for accomplishing task-level interpretation [Takahashi 92]. We have implemented a state machine approach which maintains the state of the system by tracking valid state transitions as operator commands are received. This method has the advantage of interpreting task-level commands in parallel with command verification. Illegal robot actions are easily prevented by the robust nature of this approach. The state machine was designed to run in parallel with the immersive graphics, voice recognition, and robot simulation. A 2-D array defines the state machine operation and is structured such that the current state number equals the row index and the voice command input equals the column index. Figure 3 shows the structure for each state element.

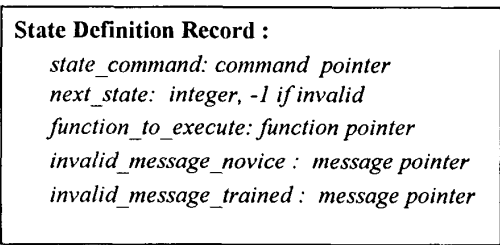


Figure 3: State Definition Record

When an input command is received, the validity of the command is verified by checking that the next\_state value is not -1. If it is -1, an invalid\_message is output

and the current state is not changed. If the command is valid, the corresponding function is executed and the current state number is update to the next\_state value. Figure 4 contains a simplified state machine diagram which shows the state transition commands. Table 1 contains the corresponding state definitions.

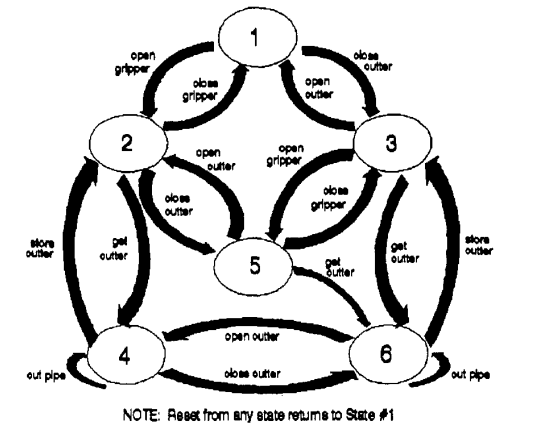


Figure 4: Simplified State Machine Diagram

State Number	Gripper State	Cutter State	Tool In Use
1	closed	open	none
2	open	open	none
3	closed	closed	none
4	in use	open	Cutter
5	open	closed	none
6	in use	closed	Cutter
7	in use	open	Cutter
8	in use	closed	Cutter

Table 1: State Definitions

The state machine starts in state 1. The state of the VR system is continuously tracked as operator commands are received. Because of the state machine configuration, command verification is immediately obtained by direct access of the state record as described above. Operator "help" is quickly obtained by searching through a row of the 2-D array where the row corresponds to the current state of the system. State configuration objects maintain the definition of each state. Thus, system "status" is obtained immediately by using the current state number as an index into the structure containing the state

configuration objects. Figure 5 shows the structure of the state objects.

**State Configuration Information:**

```
state_number : integer
gripper_open : boolean
cutter_open : boolean
gripper_in_use : boolean, T if gripper is using a tool
cutter_active : boolean, T if cutter is being used
```

**Figure 5: State Object Structure**

Overall, the state machine approach provides a very fast, robust control structure for processing, verifying and interpreting operator input sequences.

### 4.3 Robotics interface component

Control of the real ACML/UST robots from the VR environment is still under development although the communication, sensor and control subsystems are already in place.

Standard communication tools and protocols developed at Sandia will be utilized to link the VR system with the actual ACML/UST test bed. The robot subsystems of the UST/ACML test bed have been designed to respond to generically-defined commands based on the Robot Independent Programming Environment and Language (RIPE/RIPL) for autonomous systems [Miller 91]. A Sandia-developed standardized communication protocol known as GENISAS (GENeralized Interface for Supervisor and Subsystems) [Griesmeyer 92] will be used to communicate and execute the RIPE/RIPL command set on the actual ACML/UST robot system from the VR environment. The Intelligent System Operating Environment (ISOE) protocol [Griesmeyer 92] will be used for communication of robot joint vectors, sensor information and tool status from the ACML/UST workcell during command execution back to the VR system for user monitoring. Sensors on the robotic systems, in combination with low-level robot control routines, will carry out the fine robot motions. This sensor-based control reduces the accuracy requirements

on the graphical model in the VR environment since inaccuracies in graphical models cannot be avoided. With the VR controls, VR to robot communications, and robot sensors in place, the operator will be able to carry out complete robot operations from programming, to previewing and execution from the VR environment.

## 5 Conclusions and future work

The VR system described here provides an intuitive, interactive simulation interface which can be used for control and training on complex robotic systems. The task-level nature of the voice activated commands enables users to be quickly trained and allows the VR system to remain independent of the particular robot platform at the highest level. The state machine approach provides a robust control structure for quickly interpreting operator voice commands and responding with appropriate audio feedback. The VR interface allows operators to attain a comfort level with the robotic environment by immersing them in the graphical robot workcell and allowing them to move through and interact with the environment as they might with the real world. This VR system has been demonstrated to a large number of participants throughout its development. Most of the participants grasp the operation of the system quickly and their feedback has served to further improve the intuitive nature of the system.

There are several areas where the VR system discussed here will be expanded to increase the usefulness and reality of the virtual experience. A position tracker and corresponding graphical pointer are going to be integrated into the system to provide the operator with the ability to dynamically select actions and objects in the virtual world. Multi-media information, such as manuals, video sequences, schematics and still photos, will be incorporated to improve the training aspect of the system [Stansfield 93]. The monitoring of robot operations will be enhanced by including sensor data in the Robot-to-VR communication such as was done in the collision avoidance/tele-operation work by Novak, et al. [Novak 93]. Live video information from the actual robot work space may be overlaid onto the virtual model to enable the model to be updated dynamically [Cooke

93]. Force feedback would be a useful addition to increase the operator's sensory awareness of the remote robot environment and to enhance the realism of the virtual reality interface [Kim 93], as would sound. Sandia is currently carrying out research in many of these proposed extension areas. Our goal is to eventually integrate all of these technologies into a complete, interactive virtual reality system.

## 6 Acknowledgments

We would like to thank several individuals who have helped to make this project a success: Chris Cooke for contributions and assistance, and Dave Miller, Jim Tauscher and Brady Davies for their knowledge of and interface support for the ACML robotic facility. Also, sincere thanks to the many demo participants who have contributed valuable insights and enthusiasm.

## References

- [Bejczy 80] Bejczy, A. K., "Sensors, Controls, and Man-Machine Interface for Advanced Teleoperation", *Science*, Vol. 208, No. 4450, pp. 1327-1335, 1980.
- [Cooke 93] Cooke, C. A. and Stansfield, S. A., "Interactive Graphical Model Building Using Telepresence and Virtual Reality", submitted for publication to *IEEE International Conference on Robotics and Automation*, April 1994.
- [Davies 93] Davies, B. R., "Remediating Hazardous Waste Robotically Using a High-Level Control System and Real-Time Sensors", *Proc of the SPIE International Symposium on Optical Tools for Manufacturing and Advanced Automation*, September 1993.
- [Griesmeyer 92] Griesmeyer, J. M., et al., "Generic Intelligent System Control (GISC)", *Sandia Internal Report, SAND92-2159*, October 1992.
- [Helsel 91] Helsel, S. K. and Roth, J. P., Virtual Reality Theory, Practice, and Promise, Meckler Pub., 1991.
- [Kim 92] Kim, W. S. and Schenker, P., "A Teleoperation Training Simulator with Visual and Kinesthetic Force Virtual Reality", *SPIE Human Vision, Visual Processing and Digital Display III*, Vol. 1666, pp. 560-569, 1992.
- [Lozano-Pérez 89] Lozano-Pérez, T., et al., "Task-Level Planning of Pick-and-Place Robot Motions", *IEEE Computer Magazine*, pp. 21-29, 3/89.
- [McDonald 93] McDonald, M. J. and Palmquist, R. D., "Graphical Programming: On-Line Robot Simulation for Telerobotic Control", *Sandia Internal Report, SAND93-0003C*, 1993.
- [Miller 91] Miller, D. J. and Lennox, R. C., "An Object-Oriented Environment for Robot System Architectures", *Control Systems, IEEE*, Vol. 11, No. 2, February 1991.
- [Mitsubishi 92] Mitsubishi, M., et al., "A User-Friendly Manufacturing System for Hyper-Environments", *Proc of the IEEE International Conference on Robotics and Automation*, pp.25-31, 1992.
- [Novak 93] Novak, J., et al., "Collision Avoidance During Teleoperation Using Whole Arm Proximity (WHAP) Sensors Coupled to a Virtual Environment", *Proc of SPIE International Symposium on Optical Tools for Manufacturing and Advanced Automation*, 1993.
- [Stansfield 93] Stansfield, S. A., "A Computer-Based Training System Combining Virtual Reality and Multimedia", *Proc of Intelligent Computer Aided Training and Virtual Environment Tech Conf*, 1993.
- [Stark 87] Stark, L., et al., "Telerobotics: Display, control, and Communication Problems", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 1, pp. 67-75, February 1987.
- [Takahashi 92] Takahashi, T. and Ogata H., "Robotic Assembly Operation based on Task-Level Teaching in Virtual Reality", *Proc of IEEE International Conference Robotics and Automation*, pp.1083-1088, 1992.
- [Wang 93] Wang, C., and Cannon, D. J., "A Virtual End-Effector Pointing System in Point-and-Direct Robotics for Inspection of Surface Flaws Using a Neural Network Based Skeleton Transform", *Proc of IEEE International Conference Robotics and Automation*, pp. 784-795, 1993.