

Virtual Reality Based Robot Graphic Simulation and Virtual Manufacturing System

Z. J. Xu, Z. X. Zhao, M. H. Wu, J. B. LIAO and G. Y. TIAN

School of Engineering, University of Derby,
Kedleston Road, Derby, DE 22 1GB, UK

ABSTRACT

This paper reports research into an interface between industry robots and Virtual Reality systems and creates a virtual robot for robot arm simulation, control language conversion, physical communication, robot networking and teleoperation control. The paper is based on an integrated manufacturing networking software which includes a PUMA robot, a Lansing welding robot, and CNC machines. The aim of the research is to combine VR and robot technology for system testing, to establish VR based simulation workshop for defining and checking robot arm movement trajectory, eliminating the chance of potential collision. The paper also presents a system that can interpret the recognised operations into operator-level commands using task-dependent interpretation rules and a world model. A robot executes the defined task in the workspace by replicating operator's movements in the virtual workspace.

Keywords: Virtual Reality, Robotics Graphic Simulation, Off-Line Programming, Robot Command Converter.

1. INTRODUCTION

Robotics technologies have emerged and adopted for more than forty years in manufacturing industry for various purposes, from material loading/unloading, part assembly to welding and measuring. Although the task-level intelligent robot has been proposed as a new generation robot, many robots task distribution and control still rely on specialised robot command level language programming (e.g. VAL II) and showing by teach-boxes [1]. Robot is a general-purpose equipment, but the control program need to be changed when the task is updated For the traditional hardware-requested job assignment methods, this updating procedure can only happen after the previous job is terminated. This serial working mechanism is inefficient because of its hardware-oriented and domain-specific characteristic. There is a strong demand of separating job teaching and checking from the real robot equipment operation which has led to the research enthusiasm on robot system simulation in the last

two decades. Besides of industry applications, other adoption of robotics technologies in social live and scientific research is shifting to the direction such as telerobotics and robot networking. Telerobotics is a remote monitor and control system which has the ability to interact in environments that are distant, dangerous, or inaccessible by humans for special reasons [1][4]. In contrast to conventional industry robot, the primary goal of telerobotics is real-time interactive control. One of the difficulties of telerobotics implementation is the formation of the synthetic environment of the remote site.

Virtual Reality (VR) is a new human-computer interface technology that aims to provide users a computer generated immersive environment where users can "walk-in" and make real time interacting with its context and contents. The potential of using VR as a synthetic robot and other manufacturing equipment design and simulation tools have been recognised by both research community and industry users. Figure 1 shows a Virtual Manufacturing (VM) architecture where the robot simulation is the part of it. The details of the research is presented in the following sections.

2. RESEARCH BACKGROUND

Either "teach by showing" robot or traditional program control robot is working under the condition of real robot being connected, thus the desired goal point position is recorded by teaching pendant. The input program is executed by the robot itself to test the action performance. This physical equipment related manipulation is somehow tedious and risk-possessed.

After twenty years development efforts, robot simulation technologies has become more practical. Robot kinematics, dynamics, trajectory generation and control mechanism simulation and analyse software's are broadly accepted by robot manufacturers and users, e.g. INEFFABELLE, PLACE, RPI, GRASP, IGRIP and ROBCAD [4]. Various real robot-free teaching and programming environments are also available on the base

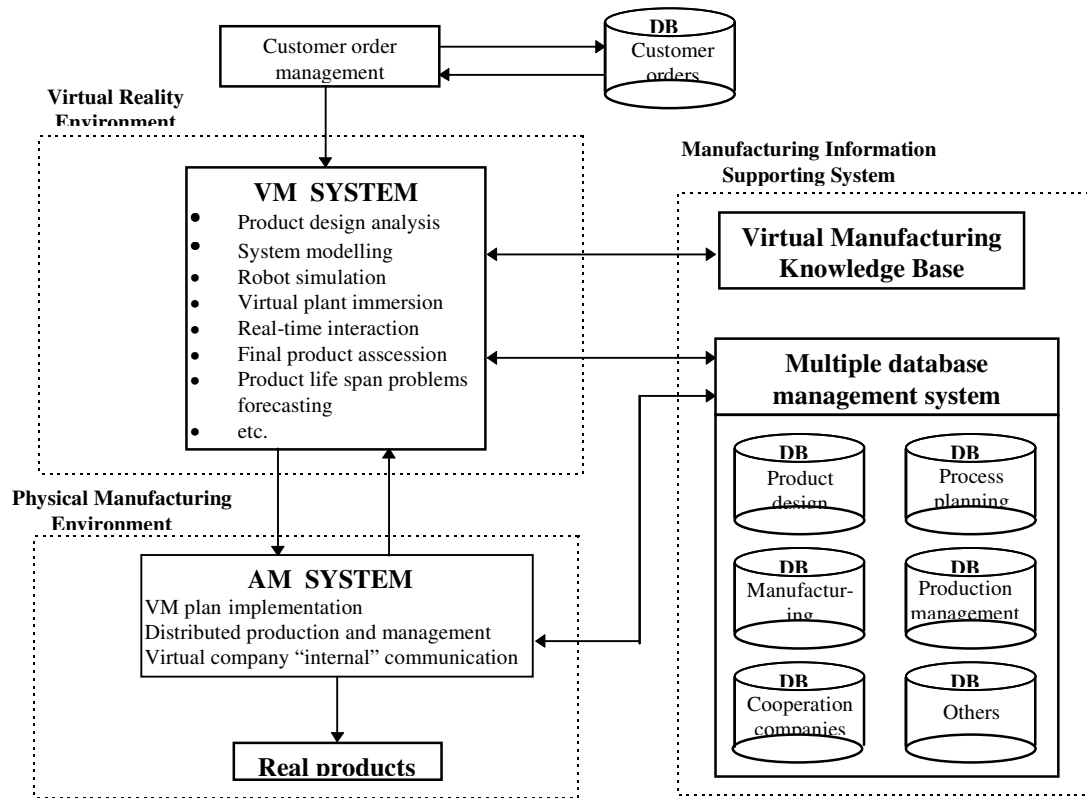


Fig - 1 The relationships between VR and VM in the prototype

of robot graphic simulation. Graphic simulation is an important feature for a complete robot simulation system. The main advantages of graphic simulation are to visualise robot system simulation results, real-time interact with computer-generated robot image, and to monitor and display feedback signals from physical robot sensory system such as vision system [2][7]. Besides basic

computer image processing functions such as translation, scaling, rotation, reflection and hidden, a robot graphic simulation system must have the enhanced characteristics of easily defining working environment, conveniently creating specific equipment, clearly describing simulation tasks, fully detecting potential collision, and 3-D animation. However, professional robot simulation system are mostly task specific software packages with restriction on other usage, for example, PLACE is mainly used on robotics workcell design and ROBSIM is for robot off-line programming, and some even has well-established standard for the graphical display devices [4].

VR based robot simulation system provides a different approach from that of robot graphical simulation. The system used in this research is a PC based VR software called Superscape VRT and SDK produced by Superscape Ltd. The advantages of using this software package include its sophisticate shape and world editor (object and environment define tools), built-in graphical interacting functions, peripheral devices communicating interface,

real-time multi-viewpoint manipulation functions (immersive and “fly”) and mathematical definition library for collision detection [3][5]. The software also provides sound editor for generating on-site 3-D audit effect. The other aim of developing VR robot simulation system is to provide a platform for the task-level intelligent robot research.

3. PROTOTYPE SYSTEM STRUCTURE AND DEVELOPING TOOLKITS

Generally, robot teaching methods have been classified in three levels, teach by showing, control by programming and task-level intelligent auto-control based on Artificial Intelligence (AI) [1]. The prototype system is designed for simulating robot control in the level of teach-playback and program control.

3.1. System structure and integrated strategy

The Robots used in this project are Unimation produced PUMA 500 assembly robot and HITACHI Co.Ltd produced welding robot LANSING PWIOII. PUMA robot use VAL II operating system, which is a computer based control system and language designed specifically for use with Unimation Industry robots. A convenient feature of VAL II is the ability to use libraries of manipulation routines. Thus complex operations may be easily and quickly programmed by combining pre-defined subtasks.

An RS-232 serial port on RUMA robot is used to communicate with the host computer. The key system functions are robot command transfer, control program (file) transfer and directly communicate Cartesian locations with real robot controller.

LANSING robot has a teaching playback system which uses a simple sequence descriptive language for controlling the operation sequence. Because there is no ready-made standard serial or parallel communication port being built in LANSING robot, a self-developed single-way communication devices is used to set-up the physical link between host computer and robot control unit, the emphasise in LANSING robot control simulation is focused on the virtual control panel simulating. The core of the established communication hardware is formed by two CMOS analogue switches 4067, which is an IC that consists of a 16-channel multiplexer, four binary control inputs and one common pin. Every combination of the input signals can determine an output channel connected with the common pin. The circuit board is designed to allow using each PC generate serial signal (8 data bits) to select two output pins (one from each chip). The two output pins are dual-way linked through the common pins so that the host computer can imitate robot control panel signals and control robot activities.

To communicate with multiple peripheral devices, an eight-port intelligent RS-232 interface board PCL-844 is installed in the host computer. All of the physical equipment are linked with the host computer through interface, includes the two robots, CNC Audit lathe and Bridgeport milling. The host computer can also connect with network sever to an local network system.

3.2. Superscape VRT and SDK

Superscape VRT is a VR studio for PCs that lets users to create real-time 3-D worlds. Users can either publish on the Internet using Viscap or display on the stand-alone Visualiser platform for DOS or Windows. VRT consists of a set of editors, e.g. World Editor, Shape Editor, Resource Editor, which users can use to create and define their own virtual world, VRT World Editor can simulate real world physical characteristics in terms of gravity, angular velocity, friction and restitution. VRT provides the Superscape Control Language (SCL) to assign behaviours to objects in the world and perform complex actions.

Superscape Developers Kit (SDK) lets users to create their own program modules that can be added into VRT and Visualise. These customised modules can perform wide range tasks from design user-specific device drives for hardware connection to transfer of Superscape data to and

from different formats. At its most basic level, the SDK consists of a C module which acts as start-up code, and header files to define functions and variables included in the Application Program Interface (API). The common methods of defining a specific virtual world is using VRT SCL or direct modification of data.

4. SOFTWARE DEVELOPMENT AND CONSIDERATION

The software development work is in three parts: (1) virtual environment simulation and definition, (2) host computer internal data, command, file management, translation and networking, (3) control software for physical equipment communication and control. The developing tool used in the first task are VRT Shape Editor, World Editor and Superscape Control Language (SCL). SCL changes objects attributes in terms of position, rotation, movement, animation, visibility, colour, lighting and bending. It also supports a wide variety of additional mathematical and programming features. Users can even connect worlds created in VRT to external programs using Dynamic Data Exchange (DDE) links to provide real-time visual representation of computer data. For the second task, an attached database management system is needed, Object Linked and Embedded (OLE) and Dynamic Data Exchange (DDE) methods are the key techniques used for connecting with external programs, Superscape Development Kit (SDK) is used to define new VRT functions, details are published in a separate paper [6]. For the third aim, a set of C source code has been programmed. It links and calls real physical system in different level at requisite time, e.g. PUMA robot can be communicated and controlled in command and file levels, however, LANSING robot can be only one way controlled in keyboard level.

4.1. VRT based robot simulation and function definitions

Every object in the VR environment is based on an orthogonal cube, VRT processes all objects according to the position and size of the bounding cube (or sorting and collision cuboids) in the virtual universe. Cubes can be related to each other and contained within a parent cube called a "group" which can be used to simplify the construction process. Each object uses a shape to define its profile. A shape consists of the points and facets that you are created in the Shape Editor, enclosed in a bounding cube. Every shape can be assigned colours, textures and movement attributes to a shape that apply to all the objects that use it. Figure 2 shows a Virtual Robot Cell with PUMA robot, Lansing robot and CNC lathe.

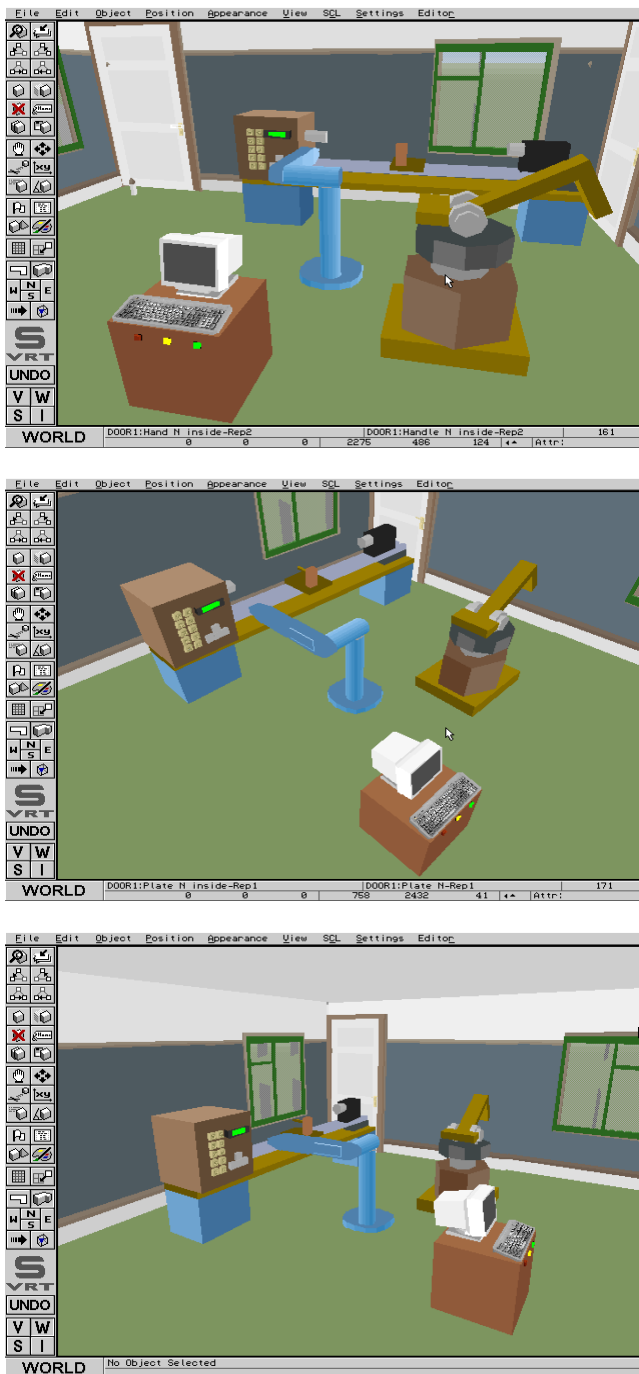


Fig - 2 Multi-viewpoint of a virtual robot workcell

The PUMA 500 robot has six degrees of freedom and all rotation joints. In the VR environment, it is constructed with 10 basic objects, e.g. robot base block (Object 1), forearm (Object 5), grip jaws (Object 9 and 10), each moveable object possessing its own SCL program (some simple joint movement can be defined directly under World Editor using Bending define functions), Each SCL program consists of a list of commands that are executed in order. A virtual robot control unit is also simulated. It includes a control computer (CRT and keyboard) and the base cabinet. The virtual computer can receive signal from

both virtual and real robot, the commands input by virtual keyboard can also control both of them. The LANSING robot simulation control is based on the self-developed communication mechanism, the working theory is stated in the previous sector. The control simulation of LANSING is focused on the keyboard simulation.

4.2. Build-up robot and host computer communication

In the PUMA robot there exists a built-in 10-pin DIN RS-232 port. The hardware linkage between the robot and the host computer is based on the serial port communication rules. The signal format is set on 9600b/s, none parity bit, 8 data bits and 1 stop bit, the pins used here are Tx, Rx, and GND. The transport sequence is followed in a dialogue mode, handshaking signals are not used. The communication software is programmed in C code. It first check the installation status of the multi-port PCL-844 board and then the pointed port. Task-specific functions with the communication board are used in the program to check the return values. If the state is correct, then the program will try to read the signals from robot control unit. If they are available, the control program will read and save them into the pre-defined host computer buffer. In the next step, users can read the message from the buffer and decide which command is to be sent out to the robot. The program is designed to support both command and file levels communication, with the ability of transferring space point records and joint arms information to other calculation programs. The LANSING robot PWIOII, produced by HITACHI in 1970s, does not have ready-made communication port with external computers. The physical hardware linkage is based on a cut-off 50-pin keyboard control port. A serial/parallel converter, a self-developed signal processing board and a switch board are thus employed to form the link. The control program is designed to recognise the users' command in the form of ASCII code, each code consisted by 8 binary bits (high-low level signal), the 4 high data bits (D7~D4) and 4 low data bits (D3~D0) being used independently to control the two IC 4067 chips on the signal processing board. The outcome is that each LANSING robot control keyboard button has a specific ASCII code to reflect. A virtual keyboard is drawn upon the outcome, the host computer input device (a mouse) is to operate the virtual keyboard in the prototype system, the LANSING robot can be driven from the virtual keyboard and demonstrate "teach-and-repeat" behaviour. In the developed software, the RS-232 port based serial communication between Audit Lathe, Bridgeport Milling and host computer are also tested.

4.3. Robot control command identification and translation

In this system, a robot control command identification and translation mechanism is used for identifying virtual robot movement, checking the generated trajectory, translating the virtual robot control programs from SCL to VAL format and reading robot feedback into a VRT

recognisable format. It also deals with the signals caught by vision, touch, or other sensory equipment.

Table - 1 shows the format of PUMA robot control program VAL II and the virtual robot control SCL program. When the program being executed, it controls the robot arm to open gripper, move to the pre-recorded space position p1 at offset along z-axis by 100mm, then reach to the point and close gripper, move back along the current z-axis of 100mm, finally, finish work and back HOME point.

VAL II Program	SCL Program
EX PROG1	if(activate(me, 0))
OPENI	moveto(x, y, z, object(9)); moveto(x, y, z, object(10));
APPRO P1 BY 100	moveby(x, y, z, object(2)); moveby(x, y, z, object(4)); moveby(x, y, z, object(5));
DELAY 0.01	wait(10000);
MOVES P1	moveby(x, y, z, object(4)); moveby(x, y, z, object(5));
DELAY 0.01	wait(10000);
CLOSEI	moveto(x, y, z, object(9)); moveto(x, y, z, object(10));
DELAY 0.01	wait(10000);
DEPART 100	moveby(x, y, z, object(4)); moveby(x, y, z, object(5));
MOVE HOME	moveby(x, y, z, object(2)); moveby(x, y, z, object(4)); moveby(x, y, z, object(5)); moveby(x, y, z, object(6));
RETURN	return(0);

Table - 1 VAL II and SCL language format

From the table, it is obvious that there is no simple one-to-one relationship between two columns, in fact each of the robot movement in the virtual environment is object-specified which is independent from other objects. This requires a considerable long SCL code and executing time to define a joint movement. The another problem is the different coordinate conversion, in the virtual robot control, there are two coordinate system used, SCL function moveto() uses the coordinate values respective to the original x, y, z axes of the virtual world, however, function moveby() uses the values relative to the object's parent. Therefore, to free real robot control unit memory and hard disk and generate the real space point coordinate from the host computer, the coordinate transfer algorithm holds a key solution. It is found that to achieve robot

command translation and coordinate conversion, SCL functions are far from sufficient, a complete set of robot system simulation functions are needed. New functions are therefore programmed and tested.

5. CONCLUSION

This paper presents a robot graphic simulation approach using Virtual Reality technology. A general VR software Superscape VRT and its user developing toolkits SDK are adopted in the prototype system. It reveals the possibility of using VR world editors to define a specific simulating environment. The devices, the built-in collision detection and coordinate translation functions provides a simplified collision detection algorithm and coordinate transition method. The developed hardware communication interface and its software robot command converter provider the feasibility of VR based telerobotics control and computer integrated Virtual Manufacturing. A WWW based networked VR world is under investigation and the idea of Internet supported Virtual Corporation forming is to be explored. However, the prototype system exposes the dilemma of keeping balance between the detail level of the simulated environment and the "reality" of it. A complete manufacturing equipment library is also proven important in manufacturing applications.

REFERENCE

1. CRAIG, J. J, 1989, Introduction to robotics: mechanics and control, ISBN 0-201-09528-9.
2. EARNSHAW, R. A., GIGANTE, M. A. and JONES, H., 1993, Virtual Reality System, ISBN 0-12-227748-1, Academic Press.
3. Superscape Company, 1996, VRT 5 User Guide and SDK Reference Manual.
4. SHUN, Z. Q., 1995, Robotic System Simulation and Its Applications, The Chinese Journal of System Simulation, Vol. 7, No. 3, September 1995.
5. VINCE, J., 1995, Virtual Reality System, ISBN 0-201-87687-6, Addison-Wesley Publishing Company.
6. XU, Z. J., Zhao, Z. X, and BAINES, R. W., 1997, Manufacturing Knowledge Acquisition and Database Management Using Virtual Reality Techniques, The 32nd International MATDOR Conference.
7. YU, J. S. and SHUN, Z. Q, 1995, Robot system graphical simulation, The Chinese Journal of System Simulation, Vol. 7, No. 3, September 1995.