

# Spatial mechanism design in virtual reality with networking

by

John Njuguna Kihonge

A thesis submitted to the graduate faculty  
in partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE

Major: Mechanical Engineering

Major Professor: Judy M. Vance

Iowa State University

Ames, Iowa

2000

Graduate College  
Iowa State University

This is to certify that the Master's thesis of  
  
John Njuguna Kihonge  
  
has met the thesis requirements of Iowa State University

---

Signatures have been redacted for privacy

Major Professor

For the Major Program

For the Graduate College

## TABLE OF CONTENTS

|   |     |
|---|-----|
| LIST OF FIGURES .....   | v   |
| LIST OF TABLES .....  | vi  |
| ACKNOWLEDGEMENTS .....  | vii |
| CHAPTER 1. INTRODUCTION .....   | 1   |
| CHAPTER 2. SPATIAL MECHANISM SYNTHESIS AND USE OF VIRTUAL<br>REALITY IN SPATIAL MECHANISM DESIGN..... | 5   |
| 2.1 Classification of Spatial Mechanisms.....   | 6   |
| 2.2 Determining Degrees of Freedom of a Mechanism.....  | 8   |
| 2.3 Synthesis of Spatial 4C Mechanisms.....   | 9   |
| 2.4 Using Virtual Reality in Mechanism Design.....  | 10  |
| CHAPTER 3. DATA COOMUNICATION AND NETWORKING .....  | 13  |
| 3.1 Internet Protocol.....  | 15  |
| 3.2 Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).....                         | 16  |
| 3.3 World2World Networking.....   | 16  |
| 3.3.1 Client / Server Architecture .....  | 17  |
| 3.3.2 Two-Tiered Server Approach and Firewall Proxy.....  | 19  |
| 3.3.3 The Server Manager and the Simulation Servers .....   | 20  |
| 3.3.4 Data Update, Transfer and Reliability .....   | 21  |
| 3.3.5 Clients and Server Synchronization.....   | 22  |
| 3.4 Trends in VR Networking.....  | 23  |
| CHAPTER 4. DESIGNING SPATIAL 4C MECHANISMS IN A VIRTUAL<br>ENVIRONMENT WITH NETWORKING.....           | 26  |

|   |    |
|---|----|
| 4.1 Virtual Environment Images and Display.....     | 26 |
| 4.2 Interaction in the Virtual Environment.....     | 27 |
| 4.3 Menus.....                                      | 29 |
| 4.4 Placing Positions.....                          | 31 |
| 4.5 Position Synthesis.....                         | 32 |
| 4.5.1 Type Map.....                                 | 32 |
| 4.5.2 Fixed and Moving Congruences.....             | 34 |
| 4.6 Testing the Mechanism and Saving to a File..... | 35 |
| 4.7 Networking and Sharing Data.....                | 35 |
| 4.8 Spatial 4C Mechanism Design Process.....        | 36 |
| CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS.....     | 41 |
| 5.1 Conclusions.....                                | 41 |
| 5.2 Recommendations for Future Work.....            | 42 |
| APPENDIX. A LINKAGE FILE SAVED FROM VRNETS.....     | 44 |
| BIBLIOGRAPHY.....                                   | 46 |

**LIST OF FIGURES**

|   |    |
|---|----|
| Figure 2.1: A Spatial 4C Mechanism.....                         | 6  |
| Figure 2.2: Iowa State University's C2 Facility .....           | 11 |
| Figure 3.1: A Simplified Communication Model.....               | 13 |
| Figure 3.2: Client /Server Configuration .....                  | 18 |
| Figure 3.3: Two-Tiered Server Approach.....                     | 20 |
| Figure 4.1: Collaboration in the C2 Facility .....              | 28 |
| Figure 4.2: Fakespace PINCH Glove.....                          | 28 |
| Figure 4.3: A Virtual Hand Model.....                           | 29 |
| Figure 4.4: VRNETS Main Menu.....                               | 30 |
| Figure 4.5: Positions Placement.....                            | 31 |
| Figure 4.6: Type Map and Color Key .....                        | 33 |
| Figure 4.7: Fixed and Moving Plane Congruences.....             | 34 |
| Figure 4.8 Program Flow Chart .....                             | 37 |
| Figure 4.9: Define a Motion Task.....                           | 38 |
| Figure 4.10: A Spatial 4C Mechanism Designed Using VRNETS ..... | 39 |

**LIST OF TABLES**

|                                 |   |
|---------------------------------|---|
| Table 2.1: Mechanism Types..... | 7 |
|---------------------------------|---|

## ACKNOWLEDGEMENTS

I would like to thank my major professor, Dr. Judy M. Vance, for the many opportunities she has given me during my stay at Iowa State University. Her help, guidance, and encouragement have been extremely valuable and will not be forgotten.

I am also grateful to my committee members Dr. Don Flugrad and Dr. Carl Bern, whose advice, support and guidance have enlightened me during my scholastic endeavors. I would also like to thank Dr. Pierre Larochelle and his team from Florida Institute of Technology for their help. Also, I want to thank all the students in Virtual Reality Applications Center, and especially those I got to work closely with, for their support.

Lastly, I would like to thank my whole family, especially my mother Mary Wanjiru Kihonge, and all my friends. Their encouragement, support and understanding have being a great source of motivation in my academic pursuits.

## CHAPTER 1. INTRODUCTION

Virtual reality (VR) can be defined as a computer-generated simulation of a three-dimensional (3D) environment, in which the user can view and manipulate the contents of the environment in an intuitive manner. A user's hand position and orientation are obtained and used to provide natural interaction with digital models. The head position and orientation are used to compute the viewing perspective for the display projected on screens. This is in contrast to the traditional human-computer interface where the user manipulates a desktop mouse and types on a keyboard to interact with digital models. Visuals, sounds, force feedback and other sensations can be used to make the virtual environment experience seem real.

VR technology is becoming more widely used and is finding users in such diverse fields as medicine, and engineering. In the medical field, for example, there is research work on VR training simulation for detecting subsurface tumors (Dinsmore et al., 1997). In engineering, VR is being used in concept development, design, prototyping, data visualization, and building and plant planning. Instead of building physical prototypes, digital models can be examined in a virtual environment using available VR software. VR is also being used in production planning, in facilities maintenance training and in marketing.

Virtual environments encourage collaboration between members of a design team. Members of the team can view the same model in real size and interact with it in an intuitive way as they discuss changes to the concept. Changes can then be easily made to the digital model and the process repeated. This replacement of physical models with digital models for

design reviews can prove to be cost effective and reduce the time it takes from concept stage to marketing of a product.

VR provides a different way to view and interact with models than the traditional human-computer interface (HCI) of a computer monitor and mouse. Models viewed using traditional HCI are not drawn in real size scale and cannot be manipulated in a natural way. VR allows the user to view the real size models and interact with the models in a very natural way through the use of a position sensor to track head motion and 3D interaction devices, such as a wand or instrumented glove.

The location and orientation of the position sensor is determined by its location and orientation in relation to a magnetic emitter placed in the virtual environment. A position sensor can be attached to the user's head so that a user can view the models in the virtual environment (VE) from different angles by simply moving around within the environment. Three dimensional interaction devices, such as Fakespace PINCH Glove, can also be equipped with a position sensor so that they can also be tracked. This ability to move around and interact with digital models naturally is referred to as immersion.

In using the traditional HCI, objects are translated and rotated in order to view them from a different angle. This may not be necessary in a virtual environment since the user can move around the objects. This makes the experience seem more real and natural than when using the traditional HCI. In addition, since the models can be drawn in real size scale in a VE, the user can perceive the effect of changes made to a model better than in a traditional HCI environment. Further, changes can be made to a digital model more cost effectively than changes performed on a physical prototype because digital models are easier to change than physical models. This leads to increased ability to investigate many what-if design scenarios.

Industries have recognized the potential of VR technology to reduce design and other costs. In the automobile industry, VR technology is being used to cut costs in the design process, improve vehicle designs, and reduce the time it takes for new designs to reach the market from the concept stage. General Motors started developing VR software for company use in 1992 (Smith et al., 2000) and at the Ford Motor Company, VR is being used in the design process before the first physical prototype is built (Stewart and Buttolo, 1999). John Deere has been collaborating with researchers at Iowa State University's Virtual Reality Applications Center to explore the use of VR in the product design process (Oliver et al., 1997).

As virtual prototyping becomes common, networking is being explored to allow participants at different sites to interact with a common database. VR networks developments will allow more collaboration between different members of a company in different geographical locations and time zones. Many organizations have manufacturing facilities in different locations, different states, or even different countries. To remain competitive, members in such organizations need to share information and models between different locations. This need has led organizations to develop collaborative tools to allow design teams from these different locations to collaborate during the design process. These networking tools will reduce duplication of tasks and will eliminate the need for frequent travel and therefore reduce costs (Smith et al., 2000).

Networking will also reduce or eliminate delays in scheduling international meetings and can lead to fast and efficient decision-making (Smith et al., 2000). What this calls for is networking where a member of a design team in the United States of America (USA) can share a model with a member of a design team in China. With a collaborative network

process, when one of the members of the design team makes a change to the digital model, the other members are able to see those changes.

The research work presented in this thesis explores the use of VR in designing spatial 4C mechanisms and the use of networking to share information between users at different locations. Spatial mechanisms allow movement in three dimensions and a spatial 4C mechanism consists of a closed linkage with four rigid links connected by four cylindrical (CCCC) joints. A cylindrical joint rotates and slides along its axis and therefore has two degrees of freedom. The design task chosen for this application is to move an object through four prescribed positions at specified orientations.

VRNETS, a software package for designing 4C spatial mechanisms with networking, was developed as part of this research. VRNETS was developed using C++ programming and uses computation routines from SPADES, a monitor-based spatial mechanism software developed at Florida Institute of Technology (Larochelle, 1998). The virtual environment is developed using the WorldToolKit (WTK) software from Engineering Animation, Inc (EAI). The networking is performed using the World2World software from EAI also.

Chapter 2 of this thesis presents an overview of spatial mechanisms and gives a brief history of spatial mechanism design in VR. Chapter 3 describes the networking process, which supports sharing of data between designers at different locations. Chapter 4 describes the program that was completed as part of this research that can be used to design a spatial 4C mechanism. Chapter 5 presents conclusions of this research and gives recommendations for future work.

## **CHAPTER 2. SPATIAL MECHANISM SYNTHESIS AND USE OF VIRTUAL REALITY IN SPATIAL MECHANISM DESIGN**

According to Erdman and Sandor (1997a), a mechanism can be defined as a linkage that consists of closed or open links connected by joints with at least one fixed link and at least two mobile links. Mechanisms are generally composed of rigid links with combinations of the following joints: revolute (R), prismatic (P), helical (H), cylindrical (C), spherical (S), or plane (P). The purpose of the mechanism, according to Erdman and Sandor (1997a), is to transfer "motion and/or force from a source to an output".

Mechanisms can be designed to perform path generation, function generation or motion generation (Erdman and Sandor, 1997a). In path generation, mechanisms are designed to move a specified point on the mechanism along a desired path. In function generation, mechanisms are designed to produce a specified function of the input motion. In motion generation, mechanisms are designed to control the position and orientation of an object attached to a mechanism.

A spatial mechanism can be designed to move a rigid body through a finite sequence of prescribed locations in space. For spatial mechanisms, combinations of rotation about three nonparallel axes and translation in three directions are possible depending on the configuration of the mechanism. Kinematics synthesis is performed to obtain a set of solution for the size and orientation of the links of the mechanism after four positions and orientations are prescribed. Kinematics is the study of motion and has been referred to as the science of relative motion. In mechanism analysis, one of the links is defined as the fixed frame of reference and all motion with respect to this link is then termed as absolute motion.

Figure 4.1 shows a spatial 4C mechanism. It is composed of four links and four cylindrical joints (CCCC) and provides motion in a 3D (spatial) space. This chapter presents an overview of spatial mechanism design and a discussion of the history of virtual reality applied to spatial mechanism design.

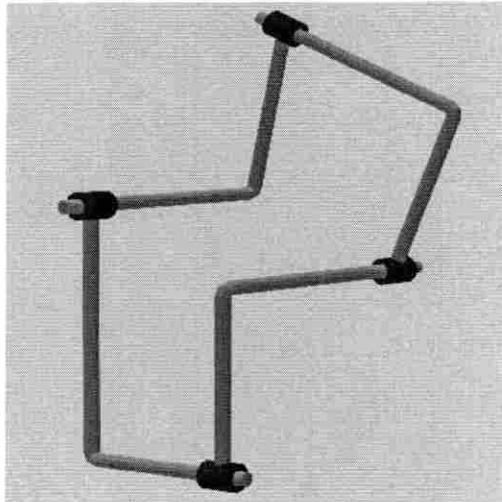


Figure 2.1: A Spatial 4C Mechanism

## 2.1 Classification of Spatial Mechanisms

The Grashof condition states that one link of the mechanism will fully rotate if the sum of the longest and the shortest links is less than (or equal) to the sum of the other two links. The Grashof condition can be extended and used to classify spherical and spatial mechanisms (Murray and Larochelle, 1998). This can be applied to spatial mechanisms because there is a spherical image associated with each spatial mechanism. The spherical image is a spherical four-bar mechanism with link lengths equal to the angular twist of the links of the spatial 4C mechanism (Duffy, 1980).

Therefore, spatial mechanisms can be classified according to the mechanism type of their corresponding spherical image. The linkage types of the spherical image are computed using the method presented in Murray and McCarthy (1995). Table 2.1 shows the different types of spherical mechanisms that are associated with spatial mechanisms.

These classifications are important because they describe the characteristics of a mechanism that a user may need to know to select the best mechanism from a set of possible solutions. Knowledge of the type of mechanism that can be selected in the design process aids in choosing a mechanism to perform a desired task.

Table 2.1: Mechanism Types

| Mechanism Types       |
|-----------------------|
| Crank-Rocker          |
| Rocker-Crank          |
| Grashof Double Rocker |
| Double Crank          |
| Non-Grashof           |
| Parallelogram         |
| Rhombus               |
| Folding Four-Bar      |
| Kite                  |
| Not a Four-Bar        |

## 2.2 Determining Degrees of Freedom of a Mechanism

Different types of joints have different degrees of freedom. Revolute, prismatic, and helical joints each have one degree of freedom. A cylindrical joint has two degrees of freedom, consisting of one rotation and one translation. A spherical joint has three degrees of freedom, consisting of three rotations. A plane joint has three degrees of freedom, consisting of two translations and one rotation.

Spatial 4C mechanisms have two degrees of freedom. The degrees of freedom of spatial mechanisms can be determined using the following equations (Erdman and Sandor, 1997a):

$$F = \lambda(l-j-1) + \sum f_i \quad (2.1)$$

Where

$F$  = degrees of freedom of the mechanism

$l$  = number of links in the mechanism, including the fixed link

$j$  = number of joints in the mechanism

$f_i$  = DOF of  $i$ th joint

$\lambda$  = DOF of the space within which the mechanism operates (for spatial mechanism  $\lambda = 6$ )

Degrees of freedom (DOF) represent the number of independent inputs required to determine the location of all links of the mechanism with respect to the fixed link. When designing a mechanism, this is important since the degrees of freedom of the mechanism indicate the number of inputs required to guide a mechanism through the prescribed positions.

### 2.3 Synthesis of Spatial 4C Mechanisms

A spatial mechanism can be designed to move a rigid body through a finite sequence of prescribed locations in space. Synthesis of spatial 4C mechanisms is based on spatial generalization of center point curves of planar kinematics and center axis and circle axis cones of spherical kinematics (Larochelle, 1995). The results of the spatial generalization are referred to as the fixed and moving congruences. These congruences are sets of lines that define the axes of CC dyads that guide a body through four prescribed positions in space. A compatible pair of fixed and moving axes maintains a constant normal distance and angle in each of the four positions of the moving body creating a rigid coupler link. The spatial triangle technique by Murray and McCarthy (1994) is employed to compute the congruences resulting in a parameterized set of lines. The moving and fixed congruences correspond to moving and fixed dyad choices.

When spatial 4C mechanisms are synthesized for motion generation, the synthesis guarantees that the mechanism can be assembled in each of the prescribed positions. This method does not guarantee that a continuous motion will take a rigid body from one location to the next because the mechanism may suffer from circuit and branch defects. According to Chase and Mirth (1993), a circuit is defined as "all possible orientations of the links which can be realized without disconnecting any of the joints". There exists more than one circuit for a mechanism if the mechanism must be disassembled and reassembled to move between two desired positions. This kind of mechanism exhibits circuit defects.

A mechanism is said to suffer from branch defects if it enters a configuration that requires an additional mechanical input to guide the moving body as desired (Larochelle, 1999). Stationary configurations will divide a single circuit into a series of branches. For the

type map representation of the four position synthesis solutions, spatial 4C mechanisms are analyzed to eliminate branch and circuit defects in motion generation tasks (Larochelle, 1999). Mechanisms that have these defects are filtered so that mechanisms without the defects are highlighted and the mechanisms with defects are darkened on the type map.

## **2.4 Using Virtual Reality in Mechanism Design**

A virtual environment is the environment that digital objects are placed in when developing a virtual reality application. The virtual environment is generated by a computer and can be audibly, haptically and/or visually displayed to the user of such an environment. VR technology displays computer information in a three-dimensional environment, taking advantage of the user's natural ability to perceive an image and its spatial characteristics.

Design and analysis of mechanisms relies on the designer's ability to visualize relative motion of the moving links. This is why the VR technology is important in the design of spatial mechanisms. Traditionally, mechanism design has concentrated on synthesis of planar motion mechanisms. Planar mechanism synthesis requires two-dimensional (2D) display and interaction, and this is well suited to the traditional HCI of a mouse and monitor. However, designing spatial mechanisms requires the designer to visualize and interact with the mechanism in 3D, which is difficult using the traditional HCI. VR allows a designer to visualize and interact in 3D when designing spatial mechanisms making it easier to specify a spatial problem in a 3D space instead of using a 2D interface.

Figure 2.2 shows Iowa State University's C2 virtual environment facility. This facility is a 12x12 feet room with stereoscopic projection screens on three walls and the floor. It has



Figure 2.2: Iowa State University's C2 Facility

a three-dimensional sound system and 3D interaction capabilities. Left and right eye images are displayed alternately on the screens. CrystalEyes shutter glasses are used to perceive images in 3D. These glasses flicker in sync with the images on the projection screens so that the user wearing the glasses sees a three dimensional environment.

Osborn and Vance (1995) developed SphereVR as the first VR environment for the design of spherical four-bar mechanisms. VR offered 3D visualization and interaction that made it intuitive to locate and interact with the positions. SphereVR had the user place coordinate frames on a sphere. The solution code for the mechanism synthesis of SphereVR was based on Suh and Radcliffe's displacement matrix method (Suh and Radcliffe, 1967).

The SphereVR spherical mechanism design program was followed by VEMECS (Virtual Environment for MECHANISM Synthesis) (Kraal, 1996). VEMECS used solution algorithms from Sphinx, a monitor based spherical mechanism design program (Ruth and McCarthy, 1997). VEMECS essentially became a VR interface to the Sphinx software. VEMECS and Sphinx were used to evaluate the effectiveness of monitor-based software applications compared to virtual reality applications (Evans et al., 1999).

Isis followed VEMECS as a design tool for spherical mechanism synthesis in VR environment (Furlong et al., 1999). Isis, like VEMECS, uses Sphinx synthesis and analysis routines. Isis provided for a more immersive experience than VEMECS due to the ability to operate Isis in Iowa State University's C2 facility. In addition, Isis allowed for the import of digital models of the surroundings and the part geometry that was the focus of the mechanism task. Isis was used to design and build a spherical mechanism that transferred recycled cans from a feeder mechanism to a storage bin.

This thesis work is another step in using VR to design spatial mechanisms. The task is to design spatial 4C mechanisms in a VR environment and to share the designed mechanism with another user through a network. VRNets software was developed for this thesis work. Four positions are prescribed and then a mechanism is synthesized to get a set of solutions for spatial 4C motion generation design task. The solution routines are the most current routines from the SPADES software developed at Florida Institute of Technology (Larochelle, 1998). SPADES software is a monitor-based spatial 4C mechanisms design software. VRNets was developed to make the design of spatial 4C mechanisms more intuitive than using the traditional HCI.

### CHAPTER 3. DATA COMMUNICATION AND NETWORKING

Communication is the exchange of data between two devices, such as the communication between a workstation and a server. Technological changes since 1970's have led to the development of integrated systems that can process and transmit all types of data and information (Stallings, 1997). According to Stallings (1997), technical-standards organizations are developing standards focussed on a public system to make all data and information sources in the world easily and uniformly accessible. A simple communication model consists of a source, a transmitter, a transmission system, a receiver and a destination (Figure 3.1).

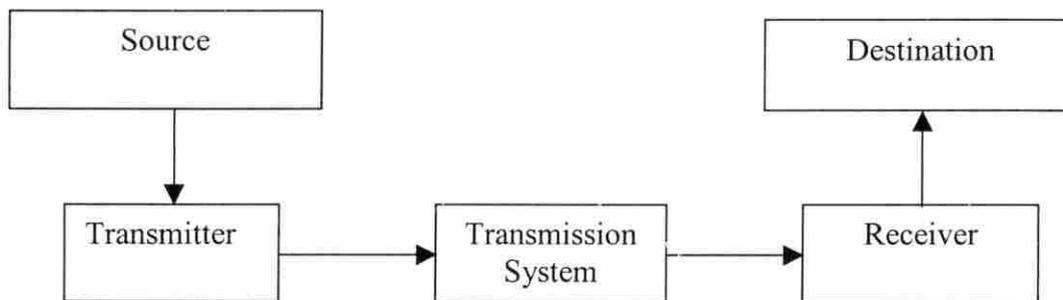


Figure 3.1: A Simplified Communication Model

A source generates the data to be transmitted while a transmitter transforms and encodes the data in such a way as to produce electromagnetic signals that can be transmitted across a transmission system. For example, a modem will transform a digital bit stream into an analog signal that can be transmitted by a telephone network - the transmission system - that connects the source and the destination. The receiver accepts a signal from the

transmission system and converts it into a form that can be handled by a destination device that takes the incoming data.

If there is a set of devices that may require a link to many others at various times and they are far apart, it is impractical to connect the devices directly. The solution is to attach the devices to a communication network, such as a Wide Area Network (WAN). The communication between devices in a network is commonly implemented using circuit switching or packet switching technologies. For a circuit-switched network, a dedicated communication is established between two stations through the nodes of a network. In a packet-switched network, data is sent in small chunks, called packets, and each packet is passed through the network from node to node along a path leading from a source to a destination.

A network is a series of points or nodes interconnected by communication paths and networking can be defined as the construction, design, and use of networks, including the physical infrastructure (Norris, 1996). Networking is also defined as the selection and use of telecommunication protocols and computer software for using and managing of the networks. As an end user, we are concerned with the later definition of networking.

The software, World2World, used in this project for networking is based on the Internet Protocol (IP). IP provides the higher layers with interface services to communicate over a physical network. Two common protocols found on the layer above the Internet protocol are the Transport Control protocol (TCP) and User Datagram Protocol (UDP) (Stallings, 1997). World2World networking software is written on top of User Datagram Protocol and it is a client / server based networking software.

### 3.1 Internet Protocol

The most widely used internetworking protocol is simply known as Internet Protocol (IP) (Stallings, 1997). IP is a method, or protocol, by which data is sent from one computer to another on the Internet. Each computer on the Internet has at least one address that uniquely identifies it from all other computers on the Internet. When data is sent or received, the message gets divided into little chunks called packets. Each of these packets contains both the sender's Internet address and the receiver's address.

The gateway computer reads the destination's address and forwards the packet to an adjacent gateway that in turn reads the destination address and forwards the packet across the internet. This process continues until one gateway recognizes the packet as belonging to a computer within its immediate domain. That gateway then forwards the packet directly to the computer whose address is specified. IP is a connectionless protocol, which means that there is no connection established between the end points that are communicating. IP just delivers the packets and it is up to another protocol, such as the Transmission Control Protocol to put them back together.

IP specifies the interface services it provides to the higher layer and the actual protocol format and mechanisms. It provides the send and deliver service primitives to the next-higher layer. The send primitive is used to request transmission of a data unit and the deliver primitive is used by the IP to notify a user of the arrival of a data unit. The two common higher-level protocols above the IP, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are described below.

### **3.2 Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)**

TCP is a connection-oriented transport-level protocol. TCP is designed to provide reliable communication between pairs of processes across a variety of reliable and unreliable networks and Internets. TCP users exchange streams of data and the data are placed in allocated buffers before they are transmitted by TCP in segments. TCP supports security and precedence labeling and provides two useful facilities for labeling data, push and urgent. TCP only uses a single type of protocol data unit, called a TCP segment.

UDP is a connectionless Internet protocol that is layered on top of IP and offers a limited amount of service when messages are exchanged between computers in a network that uses the IP (Norris, 1996). UDP is an alternative to TCP and, together with IP, is sometimes referred to as UDP/IP. UDP uses the IP to actually get a data unit, called a datagram, from one computer to another. Unlike the TCP, UDP does not provide the service of dividing a message into packets (datagrams) and reassembling them at the other end and does not provide sequencing of the packets that the data arrives in.

The application program that uses UDP must be able to make sure that the entire message has arrived and is in the right order. UDP is preferred where applications have very small data units to exchange and therefore very little message reassembling to do. UDP provides two services not provided by the IP layer. It provides port numbers to help distinguish different user requests and a checksum capability to verify that the data arrived intact.

### **3.3 World2World Networking**

World2World is a client/server based networking software for 3D real-time interactive simulations, allowing a programmer to easily develop a multi-user simulation. World2World

protocol is written on top of the UDP protocol (Engineering Animation Inc, 1998a). This avoids some of the unnecessary overhead imposed by the TCP protocol because TCP protocol acknowledgements are sent for each received packet. If no acknowledgement is received for a sent packet, the dropped packet is resent, neglecting to take into consideration that the lost packet may have already been superseded by a newer update. World2World handles the connection management, user login/logout notification, data sharing, and data grouping for a multi-user simulation.

World2World-compliant simulations work off the object/property/event paradigm used by WorldToolKit (Engineering Animation Inc, 1998b). Simulations are composed of objects and object properties. Each change made to the value of a property is known as an event. To allow multiple users to run and participate in the same simulation, each user needs to be able to receive updates made by the other users.

For example, suppose there is a graphical object in the simulation that each user can manipulate. When one user translates the object to a new location, the other user should also see the change. This is accomplished by sharing the property between the client that is modifying the property and the clients that want to receive the new property. Events cause value updates for shared properties to be sent over the network to any other clients that have also shared those properties. By limiting network data transfer to only properties that have been shared, World2World helps reduce the bandwidth usage.

### 3.3.1 Client/Server Architecture

World2World uses a client/server architecture to distribute and synchronize simulation data. A client/server approach, shown in Figure 3.2, creates a central point of

communication through which all updates are routed (Engineering Animation, 1998b) . It also provides a single point of synchronization for data consistency. A server-based system can be deployed over the Internet as well as over other local and wide area networks, unlike commonly used broadcast or multicast approaches.

In multicast or broadcast approach to networking, every machine communicates its data to the network and listens to everything on the network (Stallings, 1997). This is efficient on dedicated networks where all machines are interested in all updates. However, not all simulations run over dedicated networks. Internet is the largest example of a non-dedicated network. Over low bandwidth, forcing every participant to receive every update for every other participant severely limits the number of potential participants.

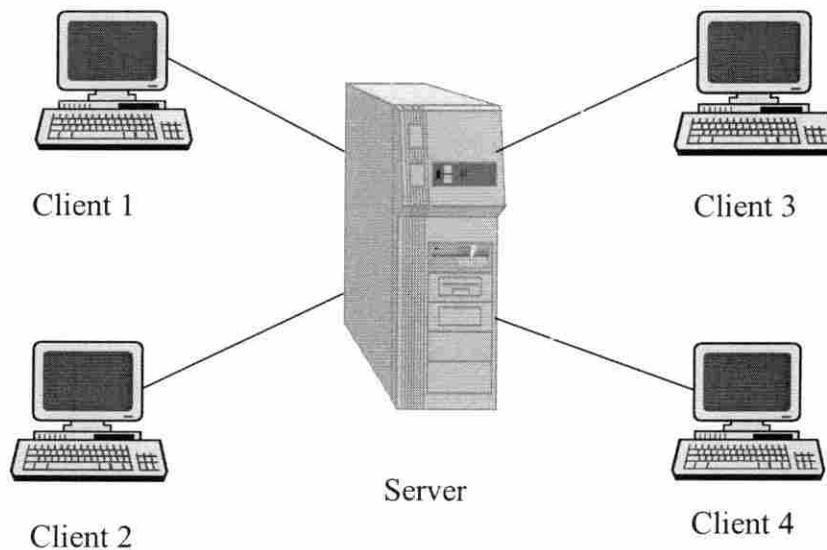


Figure 3.2 Client / Server Configuration

### 3.3.2 Two-Tiered Server Approach and Firewall Proxy

World2World uses two tiers of servers, the Server Manager and Simulation Servers, to improve scalability. Two-tiered server approach allows simultaneous processing of tasks related to both client administration and the management of shared simulation data. The Server Manager and the Simulation Server can be located on the same or on different machines (Engineering Animation, 1998b).

The servers are two different programs and can run simultaneously, making the client administration of the simulation transparent. If the Server Manager and the Simulation Server were integrated into a single program, simulations would stop until the functions performed by the Server Manager were completed. Figure 3.3 shows a two-tiered server approach with the Server Manager and the Simulation Servers located on different machines.

Firewalls serve to prevent machines outside a trusted set of machines from initiating communication with a set of machines behind the firewall. World2World Firewall Proxy is designed as a gateway for Server Managers, Simulation Servers, and clients to receive World2World data packets when they are behind a firewall. If there is a firewall between a client and the Server Manager, the World2World-compliant simulations are configured to connect to a port on the machine hosting the Firewall Proxy. The Firewall Proxy determines where to forward the clients in order to access the Server Manager based on the information provided in the World2World configuration file.

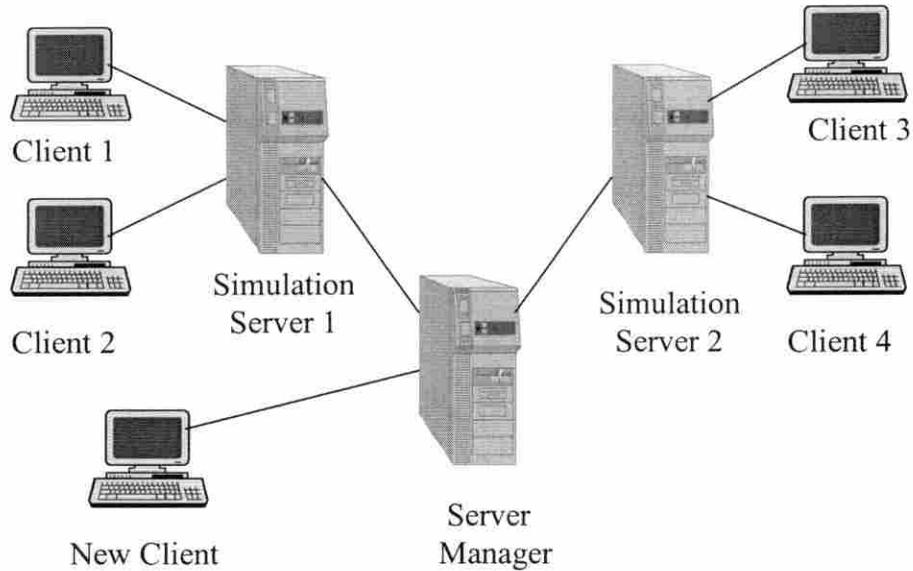


Figure 3.3: Two-Tiered Server Approach

### 3.3.3 The Server Manager and the Simulation Servers

Only one Server Manager can exist in a World2World server system, but you can have several Simulation Servers. For a new client to connect to a simulation, the client goes through the Server Manager to establish a connection and then interacts directly with the Simulation Server for the duration that the client is connected to the simulation. The Server Manager determines whether a connecting client is authorized to connect to a given simulation server. Authorization is based on the information given in the World2World configuration file and on the location of the Simulation Server that the client is interested in. If the client is authorized by the Server Manager, the Server Manager directs the client to the appropriate Simulation Server.

The Simulation Server's primary function is to store and broker access to a simulation's shared properties by maintaining the updated copy of the simulation's data.

According to *World2World User's Guide* (Engineering Animation Inc, 1998a), "Shared properties are assigned to specific *sharegroups*, providing a means of message filtering". The Simulation Server manages the distribution of the shared properties to the clients who have registered interest in the *sharegroups* that contain them, and manages any locks on shared properties and *sharegroups*.

In a single-server World2World configuration, one machine handles the workload of both the Server Manager and the Simulation Server. The processing power of such World2World server system can be improved by upgrading the speed of the machine hosting the Server Manager and Simulation Server. Another alternative would be to use multiple machines to host the Server Manager and Simulation Servers so that the workload of the Server Manager and the Simulation Server is distributed to various machines.

#### 3.3.4 Data Update, Transfer and Reliability

High volume of packets is distributed with the TCP protocol and this can significantly affect performance in a shared environment. For example, an acknowledgement is sent for each received packet and if no acknowledgement is received for a packet, the dropped packet is resent (Stallings, 1997). Since most data transmitted in networked simulations are updates for frequently updated data, resent packets may not even be used because lost data may have been superseded by a newer update.

World2World reduces packet overhead by bundling acknowledgements with other data, and resending dropped packets only when the affected updates have not been superseded by more recent data. This is accomplished by writing World2World on top of UDP and incorporating its own acknowledgement functionality. If no other data is being

sent, acknowledgements are held for one second before being sent. This helps reduce packet traffic in two ways: first, it allows more acknowledgements to be bundled, since more updates will likely have been received, and second, it increases the chance of being able to avoid resending outdated information in the event of lost packets.

The default update rate of a Simulation Server connection is equal to a client's frame rate. However, World2World allows the user to specify update frequencies for shared properties and update rates for the connection to the Simulation Server(s). The update rate for a shared property indicates how often the updates are queued up to be sent across the network. The update for a connection indicates how often packets are actually sent to the Simulation Server and vice versa. This has the potential to reduce the number of packets being sent since only the most recent update for any given property will be sent at any one time.

### 3.3.5 Clients and Server Synchronization

The system clocks on the client machines are likely to be unsynchronized and so World2World internally determines a global simulation time, which each client references. The global simulation time is derived from the estimate of time it takes for a change by a client to reach the Simulation Server. Synchronization allows a client to set the value of a property at a specified time and have that time in sync for all clients. It also allows the incorporation of dead reckoning algorithms into the simulation using the timestamps from each property change event.

### 3.4 Trends in VR Networking

As a tool, VR networking facilitates the product design process by allowing users in different locations to review the same models at the same time. This reduces the need for team members located at different geographical areas to travel to one location to review designs with other design engineers. This is a tool that can also be useful in communicating with customers and field representatives about changes to designs. Direct collaboration leads to new interaction possibilities, especially concurrent interaction of several users with one or more objects.

Distribution of the virtual environment objects complicates networking of VR environments because a large amount of information needs to be shared between two environments if all objects and their behavior were shared. In a majority of networked virtual worlds, information sharing is limited to sharing of object geometry only. However, sharing of objects can be extended to include dynamic object behaviors. Behaviors that can be shared include simple behaviors, virtual world dependent behaviors, and reactive behaviors (Singh et al., 1995). An example of a simple behavior is a linearly interpolated animation. For virtual world dependent behavior, the behaviors of the object depends on properties of the virtual world in which the object is located. In reactive behavior, the objects react to events generated by the user or other objects. BrickNet software has extended the sharing of objects to include sharing of dynamic object behavior on a network (Singh et al., 1995).

Some of the challenges in VR networking are that VR software tends to be limited in capability, is difficult to extend and may only be available on a few platforms (Watsen and Zyda, 1998). This makes it difficult to develop a networking application between different platforms. For this reason, networking applications are only developed between similar or

compatible platforms. Networking will benefit from increased use of common VR software since members in different organizations will be using the same software and can share objects in a VR environment.

General Motors (GM) developed Collaborative VisualEyes, a VR software for internal company use, in the early 1990's (Smith et al., 2000). This software was later extended for use as a networking tool using CAVERNsoft, making the sharing of distributed databases transparent to the developer of a virtual environment (Smith et al., 2000). VisualEyes is used at GM to connect several of GM's virtual environments collaboratively worldwide. The software is used for design review of shared models but the users communicate verbally through a speakerphone.

Different toolkits are available from academic and commercial institutions to facilitate the development of VR networking applications. Development of toolkits that are extensible and portable over different platforms will lead to wider use of networked VR environments. According to Watsen and Zyda (1998), Silicon Graphics Incorporated is "developing a multi-platform OpenGL-based, scene-graph, C++ API to be known as OpenGL++". Standard higher-level graphics libraries, such as OpenGL++, provide cross-platform scene-graph construction and manipulation. Using such libraries for application development instead of low-level libraries reduces the portability problem across different platforms.

Bamboo, a portable system to enable dynamically scalable virtual environments on the network, was created using the ADAPTIVE Communication Environment (Watsen and Zyda, 1998). Bamboo's architectural design mechanism supports the current Internet Protocol, IP4, and the new version, IP6. Other developments, such as the use of Virtual

Reality Modeling Language (VRML), will expand the use of VR technology to those without VR systems (Broll, 1997). VRML is becoming established as the standard for 3D data on the Internet. The World2World software used in this research is portable across Unix and Windows platforms.

## **CHAPTER 4. DESIGNING SPATIAL 4C MECHANISMS IN A VIRTUAL ENVIRONMENT WITH NETWORKING**

VRNETS has been developed at Iowa State University as a virtual reality spatial mechanism design program that allows users to network during the design process. VRNETS uses the SPADES synthesis and analysis computation routines developed at Florida Institute of Technology (Larochelle, 1998a). VRNETS is developed using C++ programming and uses Engineering Animation WorldToolKit libraries for simulation management and WorldToolKit Immersive Display Option (WTKIDO) libraries for virtual reality display management (Engineering Animation Inc, 1998b).

A user is immersed in the virtual environment using CrystalEyes shutter glasses to view stereo images in Iowa State University's C2 facility. For interaction, a Fakespace PINCH Glove is used as the primary interaction tool. Flock of Birds magnetic trackers provide position and orientation tracking and a menu system was developed to allow a user more interaction choices.

This chapter describes the virtual environment, ways of interacting with the environment and spatial 4C mechanism design process. Step-by-step procedure that a user to can follow to design a spatial 4C mechanism with networking is also presented

### **4.1 Virtual Environment Images and Display**

The management of the virtual environment display is done using the WorldToolKit Immersive Display Option. The VRNETS was designed for display in Iowa State University's C2 facility. The C2 is a 12x12 foot room where stereo images are projected on

three walls and the floor. For tracking purposes, one pair of glasses has a sensor on it and is tracked by an Ascension Flock of Birds magnetic tracker (Ascension Technology Corporation, 1993). This user's view and head orientation are used to compute the viewing perspective for display of images on the screens so that the four screens are perceived as a single environment. All the other users in the C2 will see the view of the person wearing the tracked glasses. For the other users, they get the best perspective of the environment if they are close to the user with the tracked glasses since that user's viewing perspective is the one displayed on the screens.

The C2 environment works well where collaboration with other users in a virtual environment is desired. Multiple users can be present in the C2 facility at the same time. Figure 4.1 shows two users in the C2 during the design of a spatial 4C mechanism. Because all users wear simple stereo glasses, this allows them to see both the stereo images and the other participants in the C2 environment. This allows for easy interaction among users and fosters collaboration within the VR environment.

## **4.2 Interaction in the Virtual Environment**

In VRNETS, interaction is performed using Fakespace PINCH Glove (Figure 4.2). The PINCH Glove has a conductive material placed on the finger tips, thumb and palm of the glove to register contact between a user's fingers and thumb. PINCH Gloves are more natural than a 3D mouse and they can be used for a wide range of gestures. To represent a user's hand in the virtual environment, a digital hand model is loaded (Figure 4.3). The virtual hand is animated by rearranging the joints of the model to preprogrammed configurations. The hand model animates four gestures which are used in the program to perform different tasks.



Figure 4.1: Collaboration in the C2 Facility

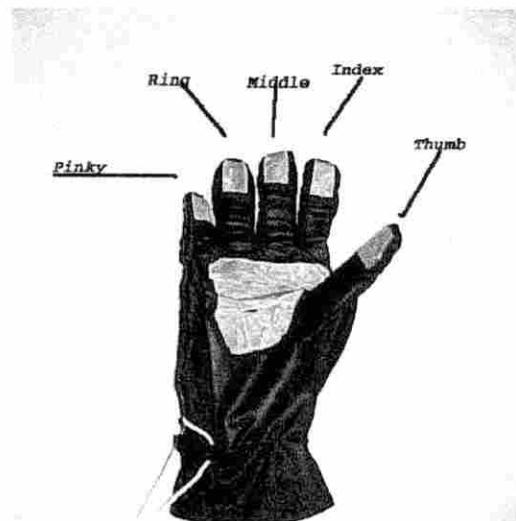


Figure 4.2: Fakespace PINCH Glove

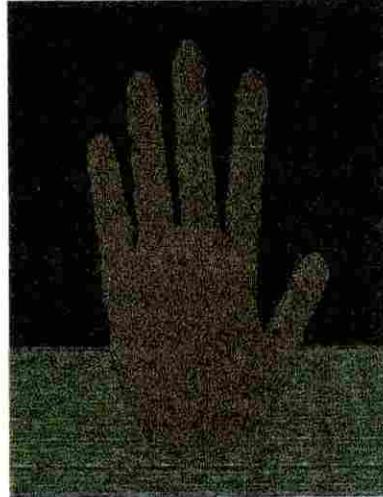


Figure 4.3 A Virtual Hand Model

The gestures are kept consistent throughout the program. The four gestures used involve contacting each of the four fingers with the thumb. For example, when the index finger contacts the thumb, there is a specific output from the PINCH Glove port, and this is programmed to produce a specific action in the program. In this program, the index finger gesture is used to grasp objects in the virtual environment. The second gesture, the thumb-to-second finger, is used to choose the menu items. The third gesture, the thumb-to-ring finger, is used to increment something in the virtual environment. The fourth gesture, the thumb-to-pinky finger, is used to open the main menu. If a user is familiar with the gestures, this offers an easy way to interact with objects in the virtual environment.

### **4.3 Menus**

Menus are used to provide more options for interaction with the VR environment. The menus are 3D objects consisting of a menu bar and text items (Figure 4.4). The main

menu can be opened at any time during the design process by contacting the pinky finger and the thumb. A menu bar can be repositioned in space by intersecting the virtual hand model with the menu bar and grasping the menu bar using the first finger and the thumb. This allows the user to move the menu to a location in the virtual environment that is convenient. A menu option is selected by intersecting the option with the second finger of the virtual hand model and contacting the second finger and the thumb. When the second finger intersects a menu option, the option is highlighted by changing the color of the text from white to green.



Figure 4.4: VRNETS Main Menu

#### 4.4 Placing Positions

To specifying the problem task, positions are placed in the desired locations and orientations using the menu and the PINCH Glove interactions (Figure 4.5). The menu system is used to load the object that will be placed in four desired positions. Once the object is loaded, it can be positioned at the first location. The orientation is changed by moving the virtual hand to the sphere on the end of each axis that is located on the object and turning the hand. In this way, orientations can be changed in each of the three axes separately. A second position is created using the incremental gesture. This position is rotated and moved to the second position. Positions three and four are similarly placed.

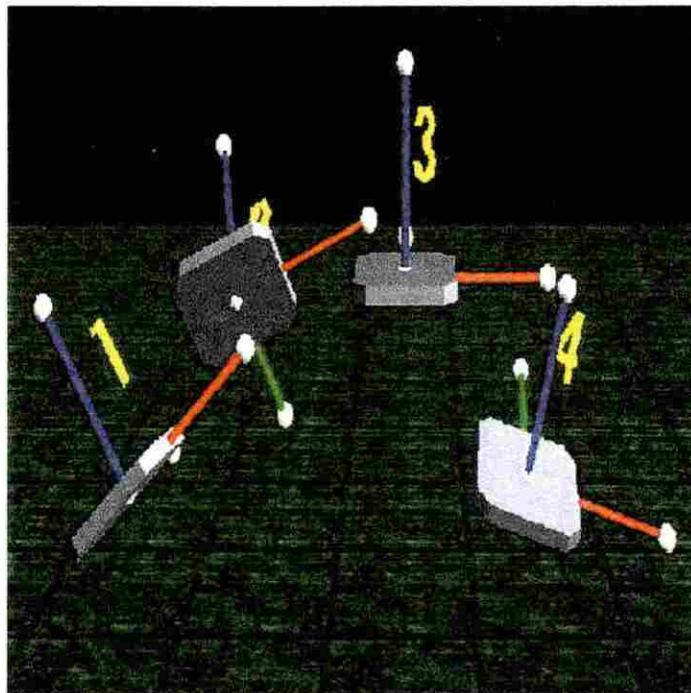


Figure 4.5: Positions Placement

After the four positions are specified and the orientations and locations are changed as desired, the incremental gesture is used to number the positions. The positions are numbered 1,2, 3 or 4 in the order the user wants the mechanism to go through the positions. After numbering, the positions cannot be translated or rotated.

## 4.5 Position Synthesis

The user synthesizes all possible mechanisms for the four positions specified and displays the results in the form desired. Either the type map or the congruences option is chosen at this point. The type map is a 2D color-coded plot of the types of the mechanism representing the solutions for the problem specified. The congruences are represented as driving and moving dyads and are drawn as yellow and red planes, respectively.

### 4.5.1 Type Map

The synthesis solution can be presented in a 2D plot referred to as a “type map”. The type map displays the solutions from the synthesis in a color-coded format showing the mechanism types (Murray and McCarthy, 1995). Spatial mechanisms are classified according to the mechanism type of their corresponding spherical image. The spherical image is a spherical four-bar mechanism with link lengths equal to the angular twist of the links of the spatial 4C mechanism (Duffy, 1980). The type map from VRNETS is shown in Figure 4.6. One axis of the map represents one choice of dyads and the other axis represents the second choice. Choosing a point on the type map is equivalent to choosing both the fixed and moving plane congruencies.

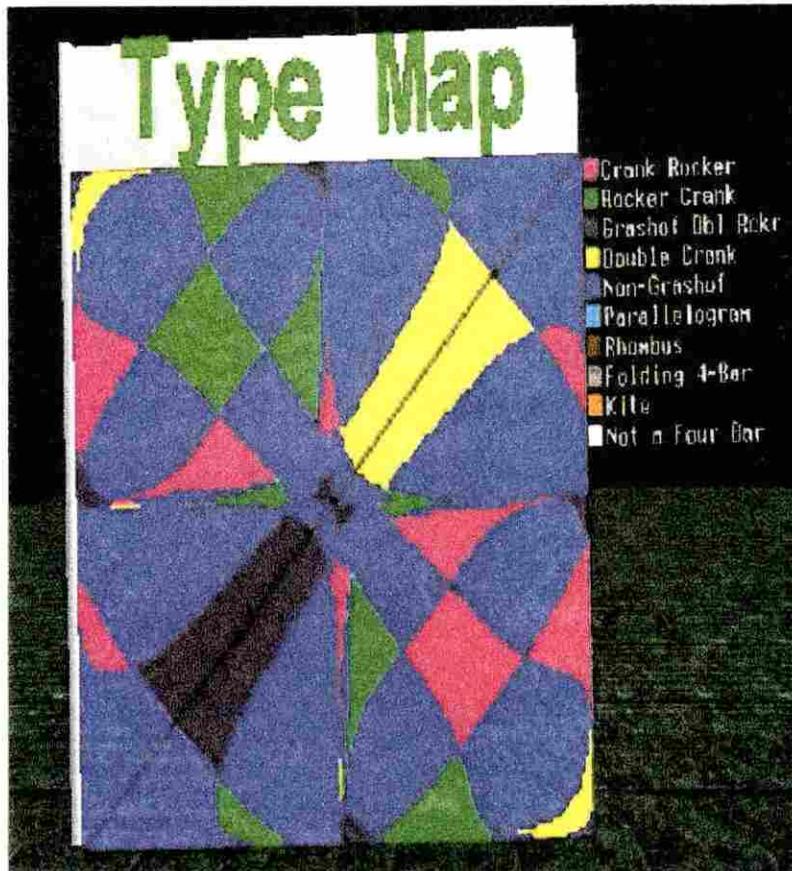


Figure 4.6: Type Map and Color Key

With the type map in the virtual environment, a pointer is drawn from the virtual hand model if the thumb and the third finger contact each other. A user selects a point on the type map by intersecting this pointer with the type map. Each point on the type map represents a mechanism solution and this will be drawn when the fingers are released. Different mechanisms can be chosen until a user gets a satisfactory mechanism.

#### 4.5.2 Fixed and Moving Congruences

The solution from the synthesis can be presented as moving plane and fixed plane congruences (Larochelle, 1995). When the congruences option is chosen, moving and/or fixed congruences can be drawn. The moving congruences represent the driving dyad choices and the fixed congruencies represent the driven dyad choices. The moving and fixed congruences are drawn in yellow and red colors, respectively (Figure 4.7). The user chooses a dyad from the congruences by moving the virtual hand and intersecting it with the congruence desired and then touching the thumb-to-second finger.

When a choice is made from the congruences, the axis of the chosen plane turns blue. A dyad is picked from the moving plane congruences and another from the fixed plane congruences to form a complete spatial 4C mechanism.

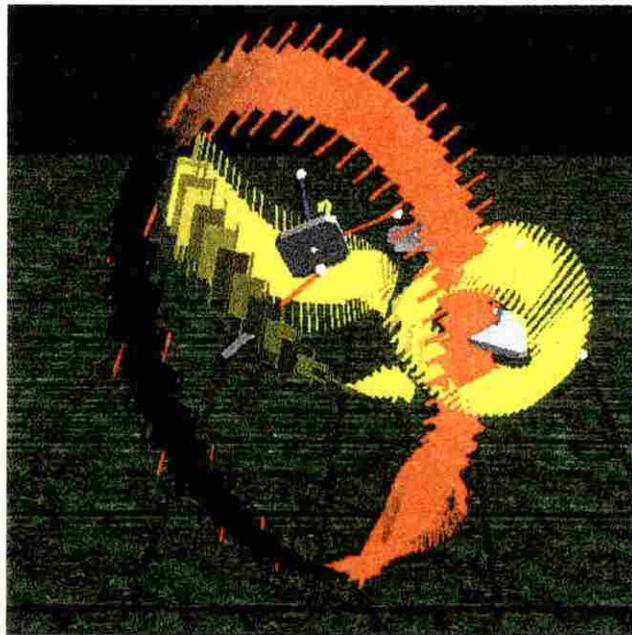


Figure 4.7: Fixed and Moving Plane Congruences

#### **4.6 Testing the Mechanism and Saving to a File**

After a mechanism has been chosen from the type map or from the congruences, it may be animated to verify it completes the task required. To complete the task, the mechanism should move the object through the four positions at the specified orientation at each position. The user can observe the motion of the mechanism to see whether the mechanism collides with objects in the virtual space and whether it goes through the positions in the required order. The user with the tracked glasses can move around the design and investigate the mechanism from different angles.

The "Main menu" has a "File" option that can be used to load an old mechanism or save a new one. After getting a desirable mechanism, an output file of the mechanism generated can be saved by selecting "Save Mechanism" option from the File menu.

#### **4.7 Networking and Sharing Data**

To run the application with networking, the user first logs on to the computer hosting the Server Manager and starts the Server Manager. Then, the user logs on to the computer hosting the Simulation Server, if it is different from the computer hosting the Server Manager, and starts the Simulation Server. After the Server Manager and Simulation Server are started, the user runs the application from any computer that can connect to the Server Manager and Simulation Server. When the application is started, it will automatically connect to the Server Manager and Simulation Server during the initialization stage. If the program is unable to connect to the Server Manager, it will exit.

In this research, VRNETS sends position translations and rotations, and mechanism link lengths, translations and rotations over the network to other users. Only one of the users

in the VR network is allowed to input the initial position information and to design a mechanism. The position information will be sent out to the simulation server and the other users will get the update for the position and the designed mechanism. The update for the position and mechanism data is done as soon as a change is made and the other users see the changes as they happen. However, the networking system speed will determine how fast the changes are received by the other users. After the mechanism has been designed, any of the users can animate the mechanism using the menu. The animation data is sent to all the users and they will see the mechanism animated in their VR environment.

Networking between two virtual environments currently requires that the two virtual environments have similar interaction devices. VRNETS was designed to allow networking between virtual environments, but was not tested between virtual environments. VRNETS networking was tested between two computers by loading a mechanism previously designed in the C2 virtual environment. Both computers displayed a monitor-based window showing the mechanism. When either of the users animated the mechanism, the other user saw the mechanism animated on his/her computer as well.

#### **4.8 Spatial 4C Mechanism Design Process**

Figure 4.8 shows a summary flow chart of the procedure used to design a spatial 4C mechanism. The gray boxes indicate the sections of the program that utilize routines from SPADES. Any of the parallel flowchart paths can be used to design a mechanism. Two options are available to design a new mechanism: first is the choice to load a base geometry

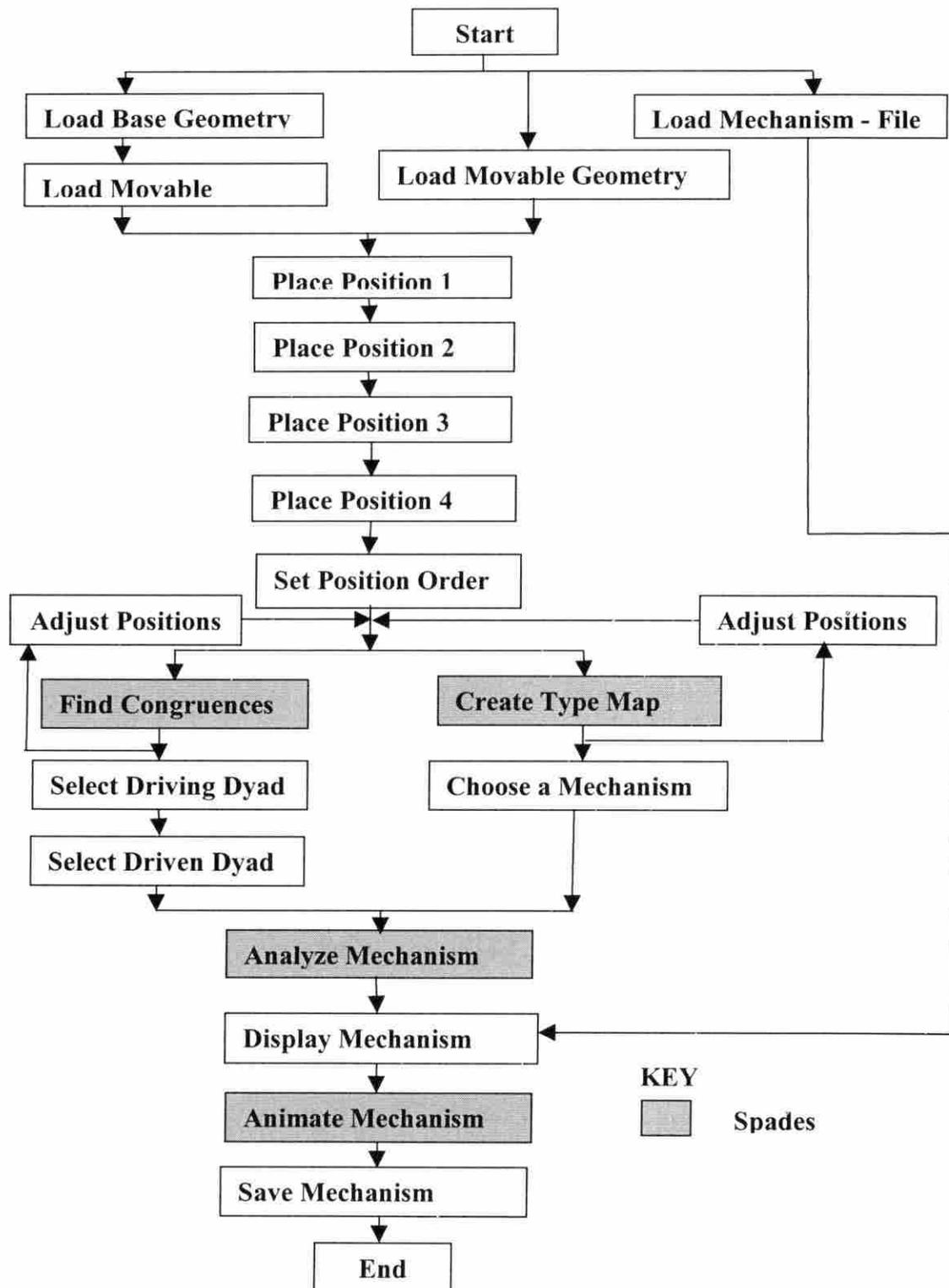


Figure 4.8 - Program Flow Chart

and then load the movable geometry and the second is to load just the movable geometry. After placing the positions and setting the order, the user can choose to find congruences or to create a type map.

This program was used to design a 4C mechanism to pass through four positions. A lathe and a table were loaded as the base geometry. The design task was to design a mechanism that will move a workpiece from the lathe to the table. Figure 4.9 shows a user in a virtual environment with the lathe and table loaded. For the task of moving a workpiece from the lathe to a table, the first and the last positions are important but the two positions in-between can be arbitrarily placed.



Figure 4.9: Define a Motion Task

Four positions were specified, with the first position being on the lathe and the fourth position being on the table. After the positions were specified, the order in which the mechanism should go through the positions was set. There were no solutions that satisfied the required task on the first attempt to design a mechanism to go through the four positions. Positions 2 and 3 were adjusted and the type map regenerated. Mechanisms were selected from the type map and animated until a satisfactory mechanism was found.

Figure 4.10 shows a spatial 4C mechanism designed using VRNETS software. The specified four positions are shown. The X-, Y- and Z-axes are drawn in red, green and blue

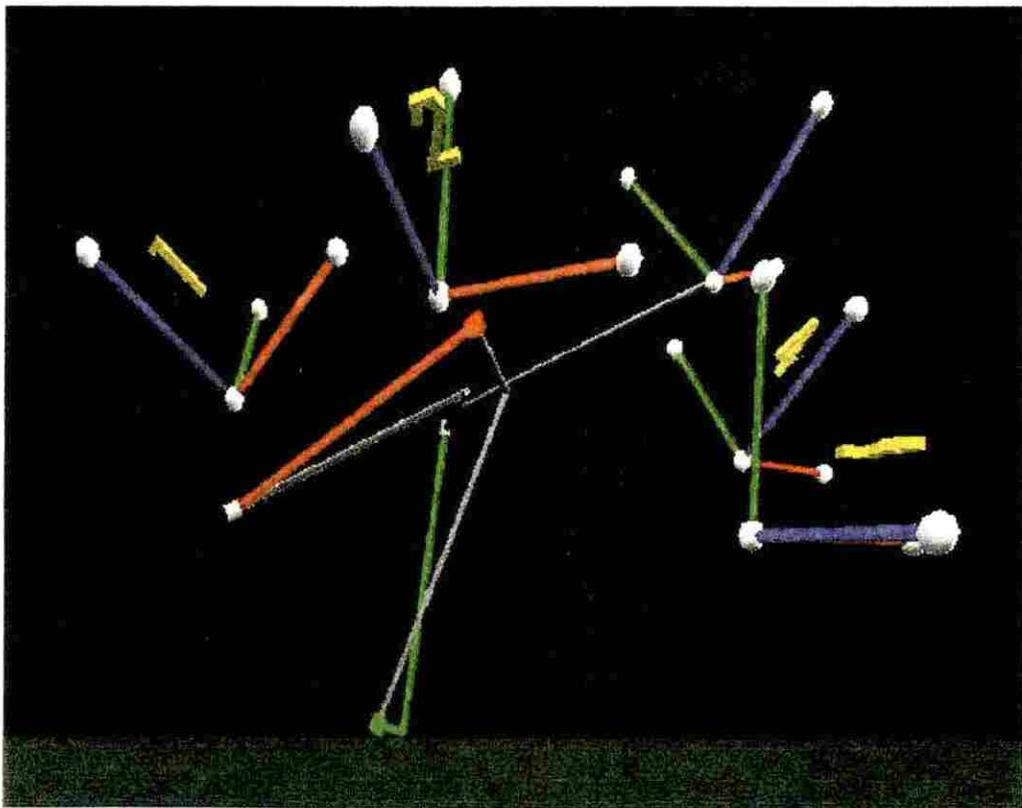


Figure 4.10: A Spatial 4C Mechanism Designed Using VRNETS

respectively in each position of the moving workpiece. The driving link is green, the driven link is red and the coupler and fixed links are gray. A coupler extension to the moving workpiece is attached to the coupler link. The coupler extension has an axes frame attached to it. This frame moves through the positions at the specified orientations during the animation to verify the motion of the mechanism. An example of a linkage file saved from VRNETS is shown in the Appendix.

In VRNETS, positions and their translations and rotations are shared so that when one user places positions in a virtual environment, users at other networked location can see the positions in their virtual environment as well. The mechanism lengths, translations and rotations are also shared. When one user designs a mechanism, the other users will see that mechanism in their environment. Changes to positions or mechanism lengths and orientation are sent to the Simulation Server and it distributes the changes to other users who are networking with that user. The software allows all the users in the network to see the positions and the mechanism and share any changes any of the users makes to the mechanism or the positions.

## CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS

### 5.1 Conclusions

Design of spatial 4C mechanism and their use will depend on the tools available for the design and verification that a mechanism accomplishes a desired task. Virtual environments allow a user to view the design environment in three dimensions and to define the task more naturally compared to traditional HCI. VRNETS was developed as a tool to design spatial 4C mechanisms in a virtual environment and to share designed digital model with other users in remote locations. In VRNETS, positions and their translations and rotations, and mechanism lengths, translations and rotations are shared.

VRNETS was implemented in Iowa State University's C2 virtual environment. VRNETS allows a user to specify a task by placing four positions in space at a desired location and orientation using the PINCH Glove and menu interactions. Solutions from position synthesis are presented as a type map or as congruences planes. A user then chooses a mechanism from either of these representations and displays it in the VE. A chosen mechanism is animated and investigated to test if the mechanism accomplishes the desired task. A user can navigate the VE by moving around naturally in the VE. VRNETS allows the user to navigate to areas outside the region confined by the C2 facility by moving the VE.

When users are networking and sharing the same digital model, for example during a design review, users at different locations view the same model. When one of the users manipulates the model, the other users viewing that model see the updated manipulated model. This allows users to discuss the design model more naturally because they are viewing the same model as if they were in the same location. The user at the alternate

location connects to a simulation server that stores the updated model data and distributes it to the users. VRNETS networking was tested between two computers by loading a mechanism that was designed in the C2 facility virtual environment.

## **5.2 Recommendations for Future Work**

VRNETS requires the user to specify four positions to synthesize the solutions. Often, only the first and last positions are critical and the intermediate positions are somewhat arbitrary. In the future, VRNETS can be improved to allow the user to only specify two or three positions. One or two more position can then be arbitrarily picked to get solutions for mechanisms that will go through the four specified positions. Research work to develop these new methods is being done at the Florida Institute of Technology and these methods can then be added to VRNETS in the future.

Another improvement that can be made to VRNETS is allowing the user to obtain more information about the mechanism when they are picking separate dyads from the congruencies. At the moment, the user picks one congruence plane and there is no indication about the type of mechanism that they will get using that congruence or which other congruence they should pick. A method can be developed to assist the user in picking the next congruence that will give a satisfactory mechanism. For example, when a user picks a driving congruence, then only those driven congruences that will combine with that driving congruence to form a mechanism without branch, circuit or order defects should be displayed. A filter could also be applied that will let the user choose the type of mechanism they want and then all the congruencies that will not form that type of mechanism would be removed.

Another possible improvement to VRNETS software would be to add the capability to do small changes in positions and orientations. Adding a system, such as a slider, to the VE can allow small changes to be made instead of having the user make changes by grasping the point and moving it. This can be useful when trying to design a mechanism to go through specified positions because small changes in orientation or in position do change the solutions obtained.

VRNETS networking was only tested between two computers by loading a mechanism that was designed in the C2 virtual environment. The software should be tested as a networking tool between two virtual environments with similar interaction devices.

The next step would be to use this software to design a mechanism for a task and build the mechanism. This would give more insight into improvements that can be made to the software. Such improvements can lead to better understanding of the design process and lead to useful applications.

**APPENDIX. A LINKAGE FILE SAVED FROM VRNETS**

Linkage\_Name: ./linkage1

Linkage\_Type: 19

Link\_Lengths(angles): 156.208 126.848 146.958 142.095

Link\_Lengths(lengths): 0.543189 1.15865 -0.0056074 1.7852

Joint\_Axis\_o: -0.45246 0.887434 0.087984 -3.51145 -1.81923 0.291578

Joint\_Axis\_a: 0.875688 -0.36052 0.321242 0 0 0

Joint\_Axis\_b: -0.909292 -0.361522 0.206132 0 0 0

Joint\_Axis\_c: 0.855928 -0.423274 -0.297028 1.2064 1.92292 0.736174

Joint\_Axis\_a in [M]: 0.548908 -0.710108 -0.440962 3.79205 1.94273 1.59183

Joint\_Axis\_b in [M]: -0.950371 -0.0772553 0.301375 -0.377717 -2.46508 -1.82302

Mode: 2

Joint\_Limits: 28.8587 -28.8587

Joint\_Angles: -154.259 -123.091 100.938 90.5215

Joint\_Distances: 4.13743 3.8804 -3.68401 -3.73556

Joint\_Angle\_Increments: -0.6 -0.0140729

Position1: 55.6551 30.8029 -36.8489 -2.46555 0.152312 0.793481

Position2: 21.9324 10.8823 2.17352e-07 -0.538063 -0.264091 -0.133599

Position3: -10.1788 -4.66124 -2.6769e-08 1.93345 1.25159 -0.385668

Position4: -26.6128 37.5692 1.07716e-06 2.98792 0.416945 1.80186 -0.310007 0.77627

0.548907 0

0.316974      0.628708      -0.710108      0

|           |            |           |   |
|-----------|------------|-----------|---|
| -0.896339 | -0.0461493 | -0.440962 | 0 |
| 4.09161   | 1.98658    | 0.276904  | 1 |
| -0.368442 | -0.096176  | -0.924663 | 0 |
| -0.812115 | -0.45079   | 0.370484  | 0 |
| -0.45246  | 0.887434   | 0.087984  | 0 |
| 0.686894  | -0.702812  | 3.88718   | 1 |

**BIBLIOGRAPHY**

- Ascension Technology Corporation, 1993, *The Flock of Birds: Installation and Operation Guide*, Burlington, VT.
- Broll, W., 1997, "Distributed Virtual Reality for Everyone – a Framework for Networked VR on the Internet", *Proceedings of Virtual Reality Annual International Symposium*, Albuquerque, NM, March 1-7, 1997, 121-128.
- Chase, T. R., Mirth, J., 1993, "Circuits and Branches of Single-degree-of-freedom Planar Mechanisms", *ASME Journal of Mechanical Design*, 115: 223-230.
- Dinsmore, M., Langrana, N., Burdea, G., Ladeji, J., 1997, "Virtual Reality Training Simulation for Palpation of Subsurface Tumors", *Proceedings of Virtual Reality Annual International Symposium*, Albuquerque, NM, March 1-7, 1997, 54-60.
- Duffy, J., 1980, *Analysis of Mechanisms and Robotic Manipulators*, Wiley and Sons, New York, NY.
- Engineering Animation Inc., 1998a, *World2World User's Guide*, Mill Valley, CA.
- Engineering Animation Inc., 1998b, *WorldToolKit User's Guide*, Mill Valley, CA.
- Erdman, A. and Sandor, G. N., 1997a, *Advanced Mechanism Design: Analysis and Synthesis, Vol. II*, Prentice Hall, Upper Saddle River, NJ.
- Erdman, A. and Sandor, G. N., 1997b, *Mechanism Design: Analysis and Synthesis, Vol. I*, 3<sup>rd</sup> Edition, Prentice Hall Inc., Upper Saddle River, NJ.
- Evans, P. T., Vance, J. M., Dark, V. J., 1999, "Assessing the Effectiveness of Traditional and Virtual Reality Interfaces in Spherical Mechanism Design", *ASME Journal of Mechanical Design*, Vol. 121: 507-514.

- Furlong, T. J., Vance, J. M., Larochele, P. M., 1999, "Spherical Mechanism Synthesis in Virtual reality", *Journal of Mechanical Design*, vol. 121:515-520.
- Kraal, J. C., 1996, "An application of virtual reality to engineering design: synthesis of spherical mechanisms", Master's Thesis, Iowa State University, Ames, IA, 1996.
- Larochele, P. M., 1999, "Branch and Circuit Analysis of the Spatial 4C Mechanisms", Florida Institute of Technology - Internal Paper, 1999, Melbourne, FL.
- Larochele, P. M., 1998, "SPADES: Software for Synthesizing Spatial 4C Mechanisms", *Proceedings of DETC'98: 1998 ASME Design Engineering Technical Conferences*, DETC98/MECH-5889, September 13-16, 1998, Atlanta, GA.
- Larochele, P. M., 1996, "Design of Spatial Mechanisms", *A tutorial sponsored by the mechanisms Committee at the ASME 1996 Design Engineering Technical Conferences*, Irvine, CA.
- Larochele, P. M., 1995, "On the Design of Spatial 4C Mechanisms for Rigid-Body Guidance Through 4 Positions", *Proceedings of the 1995 ASME Design Engineering Technical Conferences*, Boston, MA, DE-82: 825-832.
- Murray, A., and McCarthy, J., 1995, "A Linkage Type Map for Spherical 4 Position Synthesis", *Proceedings of the 1995 ASME Design Engineering Technical Conferences*, Boston, MA, DE-82: 833-838.
- Murray, A., and McCarthy, J., 1994, "Five Position Synthesis of Spatial CC Dyads", *Proceedings of the 1994 ASME Design Engineering Technical Conferences. Mechanism Synthesis and Analysis*, September 1994, DE-70: 143-152.
- Murray, A., and Larochele, P. M., 1995, "A Classification Scheme for Planar 4R, Spherical 4R, and Spatial RCCC Linkages to Facilitate Computer Animation", *Proceedings of*

*DETC'98: 1998 ASME Design Engineering Technical Conferences*, DETC98/MECH-5887, September 13-16, 1998, Atlanta, GA.

Norris, M., 1996, *Understanding Networking Technology: Concepts, Terms, and Trends*, Artech House Inc, Norwood, MA.

Oliver, J., Vance, J. M., Luecke, G., Cruz-Neira, C., 1997, "Virtual Prototyping for Concurrent Engineering", *International Projection Technology Workshop*, July 15, 1997, Stuttgart, Germany.

Osborn, S. W., Vance, J. M., 1995, "A Virtual reality Environment for Synthesizing Spherical Four-bar Mechanisms", *Proceedings of the 1995 Design Engineering Technical Conference*, Boston, MA, DE-83: 885-892, September 1995.

Ruth, D. A., McCarthy, J. M., 1997, "An Implementation of Four Position Synthesis for Planar and Spherical 4R Linkages", *Proceedings of the 1997 ASME Design Engineering Technical Conferences*, September 1997.

Singh, G., Serra, L., Png, W., Wong, A., Ng, H., 1995, "BrickNet: Sharing Object Behaviors on the Net", *Proceedings of Virtual Reality Annual International Symposium*, Research Triangle park, NC, March 11-15, 1995, 19-25.

Smith, R. S, Pawlicki, R. R., Leigh, J., Brown, D. A., 2000, "Collaborative VisualEyes", General Motors Corporation Research and Development Center - Internal Report, Warren, MI.

Stallings, W., 1997, *Data and Computer Communications*, Prentice Hall Inc, Upper Saddle River, NJ.

Stewart, P., Buttolo, P., 1999, "Putting People Power into Virtual Reality", *Mechanical Engineering Design (by ASME)*, November 1999, 18-22.

- Suh, C. H., Radcliffe, C. W., 1967, "Synthesis of Spherical Linkages With Use of the Displacement Matrix", *Journal of Engineering for Industry, Trans. ASME, Series B*, Vol. 89, 1967, 215-221.
- Watsen, K., Zyda, M., 1998, "Bamboo - A Portable System for Dynamic Extensible, Real-Time, Networked, Virtual Environments", *Proceedings of Virtual Reality Annual International Symposium*, Atlanta, GA, March 14-18, 1998, 242-249.