9-1995

A Virtual Reality Environment for Synthesizing Spherical Four-bar Mechanisms

Scott W. Osborn
Iowa State University

Judy M. Vance
Iowa State University, jmvance@iastate.edu

A Virtual Reality Environment for Synthesizing Spherical Four-bar Mechanisms

Abstract

This paper describes the development of a virtual reality environment which facilitates the design of spherical four-bar mechanisms. A short discussion of spherical mechanism design theory and computer-aided mechanism design is followed by a description of the virtual environment and the development and operation of the SphereVR program. The virtual environment allows the user to naturally interact with the input data and specify the design parameters while operating in a three-dimensional environment. We see this development as a logical extension of existing graphics-based spatial design software. The need for a three-dimensional design space is driven by the difficulty in specifying design inputs and constraints for a spatial problem using a two-dimensional interface. In addition, once the mechanism has been created, the virtual environment provides the opportunity for the user to visually verify that the mechanism will perform the desired three-dimensional motion.

Disciplines

Computer-Aided Engineering and Design

DE-Vol. 83, 1995 Design Engineering Technical Conferences
Volume 2
ASME 1995

A VIRTUAL REALITY ENVIRONMENT FOR SYNTHESIZING SPHERICAL FOUR-BAR MECHANISMS

Scott W. Osborn
Judy M. Vance[1]
Department of Mechanical Engineering
Iowa State University
Ames, Iowa

ABSTRACT

This paper describes the development of a virtual reality environment which facilitates the design of spherical four-bar mechanisms. The virtual environment allows the user to naturally interact with the input data and specify the design parameters while operating in a three-dimensional environment. We see this development as a logical extension of existing graphics-based spatial design software. The need for a three-dimensional design space is driven by the difficulty in specifying design inputs and constraints for a spatial problem using a two-dimensional interface. In addition, once the mechanism has been created, the virtual environment provides the opportunity for the user to visually verify that the mechanism will perform the desired three-dimensional motion.

INTRODUCTION

A mechanism is a machine which is used to transmit motion. It is composed of a series of links connected by various types of joints. One of the most common mechanisms is the planar 4R linkage which is illustrated in Figure 1. It consists of four links: a driver link, a coupler link, a follower link and a ground link, which are joined by four revolute (R) or pin joints, thus the 4R designation. Notice that the axes of revolution of all of the joints are parallel to each other. This mechanism transmits motion confined to a two-dimensional plane.

Mechanisms can also be assembled that provide three-dimensional motion. These are classified as spatial mechanisms. The simplest spatial mechanism is the spherical 4R which is shown in Figure 2 along with the spherical design space. It also consists of four links joined by revolute joints, however the axes of the joints

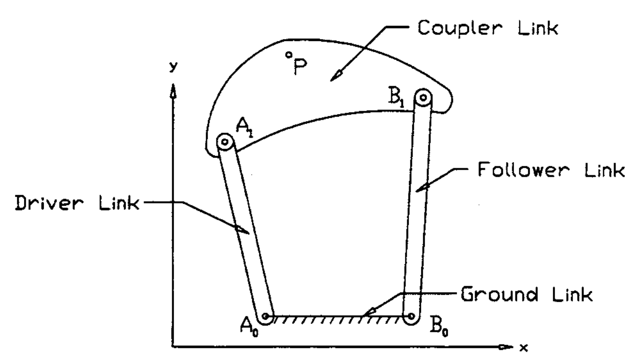

FIGURE 1: PLANAR 4R MECHANISM

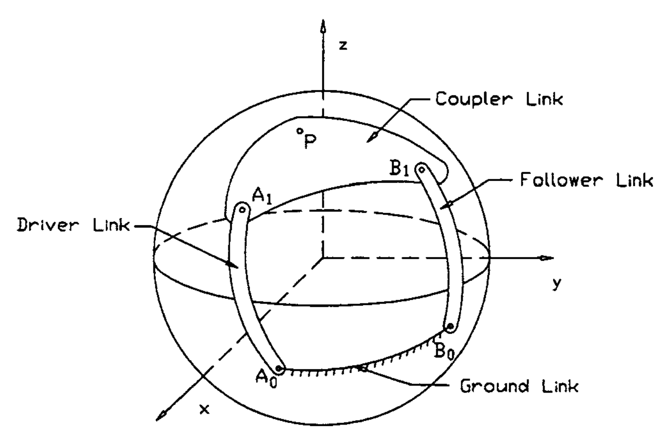

FIGURE 2: SPHERICAL 4R MECHANISM

[1] send all correspondence to this author (e-mail: jmvance@iastate.edu)

are no longer parallel, but intersect in a single point, here shown at the center of the design sphere. Because of this constraint, the moving joints, A_1 and B_1, follow paths of circular arcs about the axis intersection point. The spherical design surface shown in Figure 2 is the locus of all of the paths of the moving joints for any spherical mechanism designed with the same radius as the one shown in the figure.

Since the purpose of using a mechanism to solve a design problem is to transmit motion, the motion of each of the links is of interest to the designer. The motion of two of the links, the driver link and the follower link, provide pure rotation about one of the fixed pivots, A_0 or B_0. The ground link is fixed and therefore has no motion. It is the motion of the coupler link that can be controlled to provide both rotational and translational motion to achieve a design goal.

There are three goals driving mechanism synthesis: path generation, motion generation, and function generation. In path generation the desired path of a point on the coupler link, P, is the design specification. In function generation, a functional relationship between the input motion and the output motion is desired. In motion generation, the motion of the coupler link, as defined by the position and orientation of the link is desired. This paper deals with motion synthesis of a spherical 4R mechanism as the design objective.

COMPUTER-AIDED MECHANISM DESIGN

Computers have been used to solve complex mechanism synthesis equations for over 20 years. This was the direct result of the development of analytical synthesis methods pioneered by F. Freudenstein [1] and others. In the early 1970's, computers were used mainly for analysis of mechanisms. Programs such as IMP [2] were developed to perform planar mechanism analysis. The input to IMP was through the use of an input data file which was later processed by the IMP program in batch mode.

At about the same time, Kaufman at MIT recognized the contribution that interactive computer graphics could make to improving mechanism synthesis and developed the first interactive, graphical interface program for four-bar planar mechanism design called KINSYN [3]. In 1977, the LINCAGES [4] program (later named LINCAGES-4), another graphical interface for four-bar planar mechanism design, was developed by Erdman and Gustafson from the University of Minnesota. Later, RECSYN was developed by Waldron and Song [5].

Until the middle 1980's, graphical computer synthesis programs focused on planar mechanism design. In 1988, Thatch and Myklebust introduced a graphics workstation-based software package called Mechin [6] which could be used to design general spatial mechanisms. Mechin could be used as a graphics preprocessor for the IMP analysis program. In 1989, Chen and Erdman [7] extended LINCAGES to the design of a special type of spatial mechanism, the spherical four-bar. Currently under development by J. Michael McCarthy of the University of California at Irvine is another spherical four-bar mechanism synthesis program called SPHINX [8]. SPHINX is a menu driven, multi-windowed, UNIX-based software program.

Even though several of these graphics-based mechanism synthesis programs have been available for years, they are not widely used in industry. Even as late as 1988, Thoreson and Erdman [9] contend that most machinery designers are reluctant to embrace computer graphics-based mechanism design software and continue to synthesize mechanisms using cardboard and thumbtack models. Barris, et.al. [10] believe that the reason for this reluctance is due to the inadequacies of the user interface. This is especially significant for the case of spatial mechanism design.

The goal of spatial mechanism design is to produce three-dimensional motion. Visualizing this motion on a two-dimensional screen is difficult even using current computer graphics techniques. It is especially difficult to specify input parameters, such as the location of the pivots and the positions and orientations of the coupler link, while trying to visualize the three-dimensional design space. In addition, while planar mechanism motion is are easily visualized, the three-dimensional nature of spatial mechanisms leads to three-dimensional motion that is very difficult to visualize. Once the input parameters are specified and the mechanism is synthesized, it is difficult to verify if the resultant spatial mechanism will provide the desired motion without building a prototype. Virtual reality provides the opportunity to design in a three-dimensional design space.

VIRTUAL REALITY

Immersion in the computer generated world and navigation through the world are what distinguishes virtual reality (VR) from workstation-based three-dimensional graphics [11]. The immersivity that is integral to VR provides a computer-generated three-dimensional space for visualizing objects and performing tasks. Immersion is achieved when the user believes that he/she inhabits the same space as the computer image. The computer-generated objects occupy locations in the environment and can be manipulated like real objects. Natural head and body motions control the image projected to the user. As it relates to mechanism design, users can specify joint locations and link motion by "walking around" the design sphere and placing objects on the sphere. This is in contrast to the traditional menu-based graphics where the 2D mouse is used to rotate the sphere in order to see the other side.

The visual sensory system provides one of the most powerful cues to creating a sense of immersion. It follows that visual VR displays are the most highly developed sensory interfaces. Current technology includes several different virtual visual interface devices that provide varying degrees of immersion. Stereo shutter glasses, head mounted displays, head coupled displays, and stereo projection rooms each provide a different degree of immersion in the virtual environment. Instrumented gloves, 3D mice, and 3D wands are some of the interaction and navigation devices currently used.

Just as computer-based design has progressed from 2D CAD to solid modeling to parametric three-dimensional modeling, VR can be seen as an extension to that evolution. VR provides yet one more dimension to the design environment: the ability to use natural human senses and movements to interact with computer-generated designs.

The remaining sections of this paper describe the development of a virtual interface for spherical mechanism design.

SPHERICAL FOUR-BAR MECHANISM SYNTHESIS

A spherical four-bar is defined as a three-dimensional, one degree of freedom mechanism consisting of four links connected with revolute joints whose axes of rotation intersect at one point (see Figure 2). The four links which comprise the mechanism are the driver link, the coupler link, the follower link and the fixed link. Both the driver link and the follower link have one fixed pivot and one moving pivot. The coupler link consists of two moving pivots and a precision point. It is generally the motion of the coupler link or the precision point that is of interest to the designer. The purpose of this paper is to describe the virtual reality software that has been developed as an interface for motion generation for spherical four-bar mechanism design.

Motion generation synthesis involves defining a mechanism which controls the movement of one of the links through a prescribed motion sequence [12]. The objective of mechanism synthesis for motion generation is to move the coupler link through several specified positions in space and maintain a given link orientation at each position. Suh and Radcliffe [13] developed a displacement matrix method, based on rigid body motion, to accomplish motion generation for the spherical four-bar mechanism. This method provides the mathematical underpinnings for the virtual reality design environment presented here. A short discussion of the underlying synthesis equations follows.

One of the more difficult aspects of spherical mechanism design for motion synthesis is the specification of initial design parameters. The orientation and position of the coupler link is initially determined as a result of the design objectives. The position of the two ground pivots, A_0 and B_0, the two moving pivots, A_1 and B_1, and the precision point, P, must be determined in order to fully define the mechanism.

The displacement matrix method uses a dyadic approach to synthesize the mechanism. The first step is to divide the four-bar mechanism into two dyads connected by the precision point P (see Figure 3). One dyad contains the driver link and part of the coupler link and the other dyad contains the follower link and the other part of the coupler link. The position of the ground pivot and the moving pivots of each dyad can then be obtained independently. Therefore, one dyad will be completely synthesized before the other one is designed. The final design consists of the joining of the two dyads, where the coupler link is created by connecting one moving pivot from each dyad, A_1, B_1, and the precision point P.

For a single dyad, there are six unknowns that make up the location of the two joints: (x_o, y_o, z_o) for the fixed pivot and (x_1, y_1, z_1) for the moving pivot. Two equations are necessary to ensure that the joints stay on the surface of the sphere. These equations are

$$x_0^2 + y_0^2 + z_0^2 = R^2 \qquad (1)$$

$$x_1^2 + y_1^2 + z_1^2 = R^2 \qquad (2)$$

where R is the radius of the sphere.

Equations are needed that will keep the link length constant for all positions. These equations are

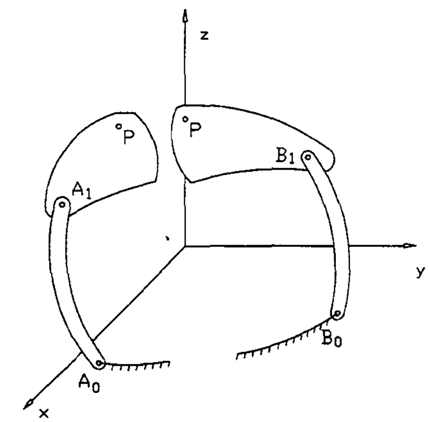

FIGURE 3: DYADS OF THE SPHERICAL 4R

$$(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2 = (x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2 \qquad (3)$$

$$\text{for } \{i = 2, n\} \text{ and } (n < 5)$$

where i is the position number of the moving pivot joint, starting at i=2. This formulation is for up to five positions of the mechanism.

In order to solve the set of six equations, the displacement matrix method developed by Suh and Radcliffe [13] is used to obtain a relationship between the location of the moving joint in positions 2 through 5 (x_i, y_i, z_i) with respect to the initial position of the moving pivot (x_1, y_1, z_1). This is possible because the moving pivot is a single point on the rigid body coupler link and the motion of the coupler link has been specified. A displacement matrix, D_{1n}, for each i^{th} position is created from the specified motion of the coupler link.

$$\begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} = \begin{bmatrix} D_{1n} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} \qquad (4)$$

This relationship makes it possible to solve for all positions of the moving pivot once the first position is known. This leaves six unknowns, $x_o, y_o, z_o, x_1, y_1, z_1$. Therefore for a unique solution, six equations are needed, which in turn means that a maximum of five positions can be specified. If fewer than five positions of the coupler are supplied, some of the unknowns must be specified.

Since the equations are non-linear, a Newton-Raphson algorithm is used to solve equations (1), (2), and (3). Once the solution has been determined for one dyad, the solution proceeds for the other dyad.

THE VIRTUAL ENVIRONMENT

The virtual reality interface consists of software which creates the graphics as well as hardware which is used to track head and hand movements. The SphereVR program was developed using WorldToolKit (Sense8 Corporation, Sausalito, CA) virtual reality software. WorldToolKit is a collection of C functions that are used to manage the sensors, peripheral devices, and computer images that are controlled by user interaction.

The BOOM3C (Fakespace Inc., Menlo Park, CA) is used to display as well as track the user's viewpoint in the virtual world (see Figure 4). The BOOM3C is a full-color head-coupled display where the visual device is suspended from a counter-weighted cantilever beam supported from a floor base. The BOOM3C resembles a robot with a monitor suspended where the end effector is usually located. The user is free to move the monitor into any position within a 3 foot radius of the base of the BOOM. The position sensing is accomplished through the use of optical encoders in each joint of the mechanism. This location information is passed to the computer to control the viewpoint such that the computer image changes according to the location and orientation of the BOOM.

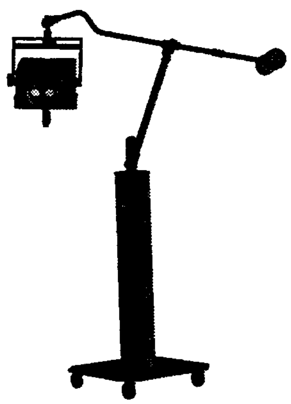

FIGURE 4: FAKESPACE BOOM

In order to interact with the virtual environment, the designer must be able to control a virtual hand or manipulator in the graphical environment. A 3D mouse (Logitech, Inc., Fremont, CA) is used in this application. The 3D mouse is similar to the 2D mouse except its use is not restricted to the table top, but it can be moved in three-dimensional space. The mouse has an ultrasonic tracker which sends signals corresponding to the position and orientation of the mouse to a receiver located on the computer.

All of the peripheral equipment is connected to a Silicon Graphics Onyx Reality Engine^2. This computer is especially designed to display complex graphical images very rapidly.

This program can easily be implemented using a wide variety of VR equipment. The equipment chosen here is fairly expensive and it will be the objective of future work to determine what degree of immersion, visual display resolution and natural interaction is necessary to implement a spatial mechanism design program, as these choices present cost trade-offs to the user. The next section will describe the program that was developed to synthesize motion generation of a spherical four-bar mechanism.

PROGRAM OPERATION

There are six major steps in the synthesis of spherical four-bar mechanisms presented by this program:

1. Specify the location and radius of the great sphere (spherical constraint surface).
2. Specify the position and location of the coupler link in up to five positions on the sphere.
3. Supply any necessary initial parameters, such as the location of ground or moving pivots, and supply initial guesses for the unknowns, in order to provide enough information for a unique solution.
4. Solve the synthesis equations and generate the mechanism.
5. Animate the linkage to verify that the solution satisfies the design objective.
6. Redesign.

To begin, the user steps up to the BOOM and places the display device up to his/her eyes. With one hand, the BOOM is held in place while the other hand holds the 3D mouse.

Step 1: Specification of the great sphere

When the program begins, the user sees a large yellow sphere (called the great sphere) that serves as the constraint surface of the mechanism. The size of the great sphere can be interactively changed by moving any of the four small red spheres which define the diameter and location of the great sphere. These small red spheres are attached to the surface of the great sphere and can be moved by using the 3D mouse. The cursor in the virtual environment is shown as a small cone. As the small spheres are moved, the great sphere is continually being drawn on the screen.

Step 2: Specification of the position and location of the coupler link in several positions

In the initial environment are five red arrowheads which are used to represent the position and orientation of the coupler link. A three-dimensional axis locator that indicates the direction of the x, y, and z axes is also in the environment. To specify the position and orientation of the coupler link in one position, the user reaches out with the 3D mouse to select an arrowhead and move it in contact with the sphere. When the arrowhead comes in contact with the surface, it is immediately attached tangential to the sphere and the viewpoint changes from a global perspective to a view closer to the arrowhead. The arrowhead objects are numbered based on the order in which they are applied to the sphere. Once attached, the arrowhead can be reoriented and relocated but it is always confined to the surface of the sphere. This procedure is repeated until one arrowhead is located for each of the specified positions of the coupler link up to a maximum of five positions. The location and orientation of the coupler link in these positions are the design objectives for motion generation synthesis.

Step 3: Specification of initial parameters and initial guesses

One of the more critical steps in the design process comes in determining initial design parameters. In this method, all of the design parameter specification is done using the mouse while looking through the BOOM. The user occupies space in the virtual environment throughout the design process. This allows the user a unique view into the evolution of the design specifications. An infinite number of viewing angles are possible to view the constraint surface of the sphere and the location of the other design specifications.

After the coupler position arrows have been placed, the user must specify the initial placement of the four joints on the sphere. Joint objects appear on the screen with the fixed pivot represented by a triangular object and the moving pivot represented by a square object. The pivots for dyad A are colored magenta and the pivots for dyad B are colored green. The pivots can be moved to any position on the sphere by using the 3D mouse.

The control panel is a key member of the virtual environment. It informs the user what needs to be accomplished in order to perform the synthesis. The panel is a large tabular object that exists in three-space and consists of twelve buttons representing the coordinates of the four unknown joints. Each button on the control panel represents a joint coordinate, with A_0 and A_1 corresponding to the fixed and moving pivot of one dyad and B_0 and B_1 corresponding to the other dyad. A message board is displayed in the middle of the control panel that displays messages to the designer to guide the design process.

Once the joints are located on the sphere, joint constraints can be applied using the control panel. For example, if the user moves the indicator cone in contact with the button corresponding to A_0 "Y" and presses the mouse button, the control panel button is depressed. This indicates that the y position of the fixed pivot of dyad A, represented by the y coordinate of the magenta triangle object in the virtual world, is a fixed constraint and will be input to the solver as an initial constraint. Specifying a joint coordinate results in specifying a constraint surface which is represented by a plane perpendicular to the coordinate axis. Because the joint is already constrained to the sphere, the result is that the joint is specified to lie on a circle of points on the sphere that share that same y coordinate.

Figure 5 shows several constraint circles corresponding to the appropriate buttons on the control panel. In this example, A_0 has a constraint on the y coordinate, A_1 has a constraint on the x coordinate, and both B_0 and B_1 have the z coordinate constrained. The color of the constraint circle corresponds to the color of the corresponding axis in the three-dimensional axis object. The three specified positions of the coupler link are also shown in the figure as arrows on the sphere.

Step 4: Solving the synthesis equations and generating the mechanism.

Once the correct number of joint coordinates and coupler positions are specified, the user is prompted by the control panel to solve. By clicking onto the "solve" message on the control panel, a solution is attempted. If successful, the mechanism will appear on the screen and the user will be prompted to begin the animation. If the algorithm failed to converge then the user is prompted to either change the constraints or move one or more of the pivot points and try again.

Step 5: Animating the linkage to verify that the solution satisfies the design objective.

Even after the mechanism has been designed, there is no guarantee that the it will continuously generate the motion that is prescribed. The synthesis only ensures that the link length is constant, that the coupler link is oriented correctly, and that the ground pivots are in the same spot as the precision points. Animation of the mechanism through it's defined position to verify the motion is necessary to complete the design.

Once the mechanism has been generated, the "animate" message appears on the control panel and the user can click onto the message and watch the mechanism animate through the prescribed motion. The path of the coupler point is traced onto the sphere (Figure 6). The user can "walk around" the mechanism to view it from any position while the animation is being performed.

Step 6: Redesign

The mechanism can be altered in real-time by interactively changing the parameters that were used to design it. The number of positions can be altered, the specified joint variables changed, and the initial guesses can be modified. This interaction is enhanced by the fact that the mechanism can be designed and viewed from an infinite number of viewpoints, including those from inside the sphere.

Once an acceptable mechanism has been created, an output file can be generated which lists the size of the great sphere, the orientation and location of the coupler link in the desired positions, the constraints applied to the pivots and the final positions of all four pivot joints. This data can be used to create a physical prototype of the mechanism or be retained to be read into the virtual reality program if further modifications are desired.

CREATION OF THE VIRTUAL ENVIRONMENT

Several programming challenges were met in the process of developing this interface and the approaches to generating the virtual environment will now be presented.

Most of the current virtual reality applications deal with objects with fixed geometry that exist in a database. Architectural walkthroughs, virtual galleries, virtual kitchen designs, etc., all deal with fixed geometry objects. These objects are generally created in a CAD program and then entered into the virtual environment. The World Tool Kit software uses the neutral file format (NFF) as the data base format for objects. The NFF file is an ASCII file which contains the essential geometry of rigid body objects: the location of the vertices, the connectivity of the vertices (polygons), the color, and any texture that is applied to a particular polygon.

Fixed geometry objects in this program include the small spheres, pivots, arrowheads, control panel and buttons, locator cone, and the axis object. With the exception of the spheres, these objects have simple geometry and therefore it was easy to design them on graph paper and write the NFF file directly. The polygonal

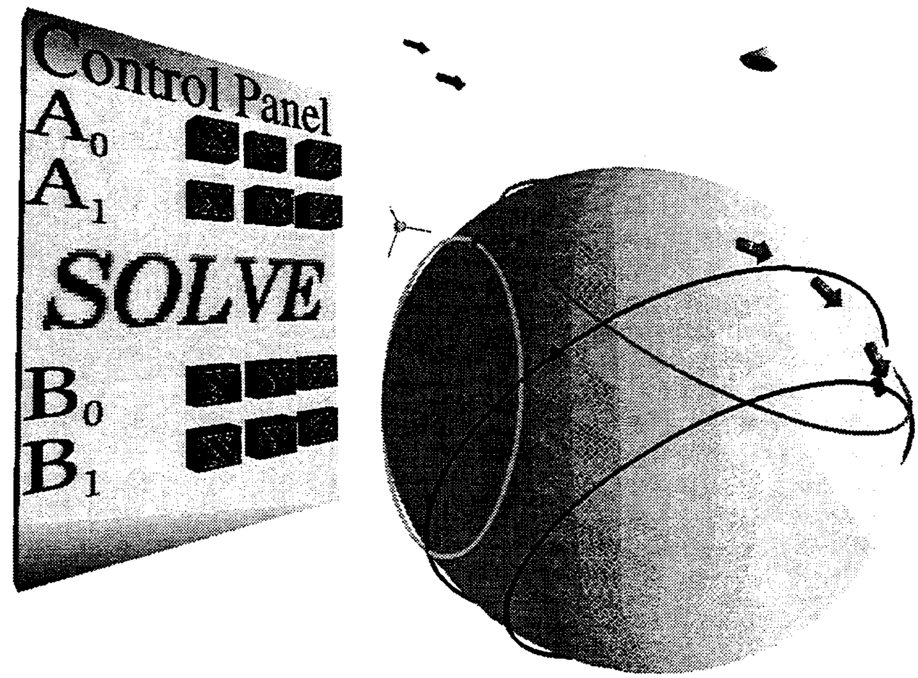

FIGURE 5: CONSTRAINT CIRCLES AND CONTROL PANEL

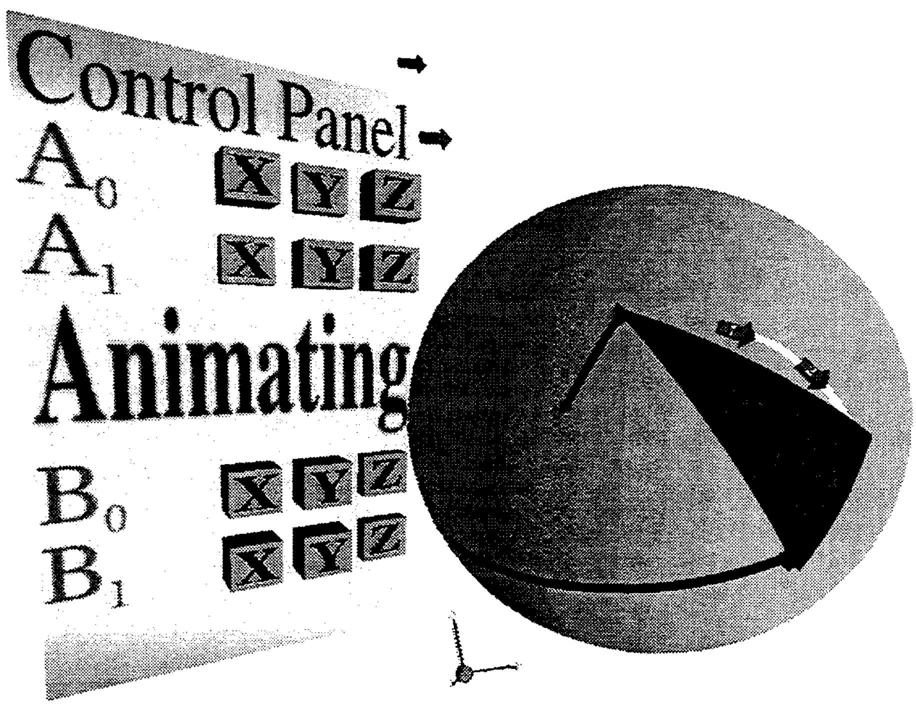

FIGURE 6: FINAL DESIGN AND VERIFICATION OF MOTION

mesh that defines the surface of the sphere was obtained from locally developed software and the data file converted into an NFF file. The mesh is structured so that it provides nearly an equally dense number of polygons on the surface of the sphere in all locations. In other words, the mesh is not uniform, but adjusted to the circumferential area of the local surface of the sphere. This eliminates the high concentration of polygons at the poles of the sphere when a uniform mesh is applied. The maximum number of equally spaced vertices representing the small locator spheres must be large enough to accurately represent the surface of the sphere, yet few enough so that the frame rate is not compromised which inhibits the immersive effects of the virtual experience. The great sphere is also defined by an NFF file, but can be scaled interactively as the program executes in order to match the defining locations of the smaller spheres.

The remaining objects in the simulation are the links that define the mechanism; the driver, follower and coupler link. These objectshave fixed geometry but the geometry changes for each new solution. Therefore these objects cannot be created prior to execution of the program and stored as NFF files. They must be created as the program operates. This is one of the unique features of this virtual environment.

The driver and follower links are defined as the connection between the fixed and moving pivot of a particular dyad. For the mechanism to perform properly, curved links are required because the links must traverse across the surface of the great sphere. Each link is created by first identifying the arc segment that lies on the surface of the sphere which connects a set of fixed and moving pivots. Since the mechanism must travel over the sphere and not within the surface of the sphere, the arc segment is offset slightly above the surface of the sphere and extended to connect to the pivot axes. The pivot axes are the lines which connect a pivot to the sphere center. The links have both thickness and width. Therefore to create the thickness, another arc segment is created an offset distance above the first segment. Each arc segment is then replaced by a strip running parallel to the surface of the sphere which has width that is proportional to the radius of the great sphere. The strips are made up of vertices that were derived by subdividing the arc segments. Both edges of the arc strips are connected by polygons to create the sides of the driver and follower links.

The coupler link consists of a triangular patch covering a region of the sphere and is created in manner similar to the driver and follower links. Unlike the driver and follower links which connect two points, the coupler link is the rigid connection between the moving pivots of both dyads as well as the precision point. The path that is traced by the precision point is created by drawing an asterisk shape on the sphere at a fixed interval as the mechanism is animated.

The control panel is an innovative approach to "bookkeeping" for the designer. The control panel keeps track of how many constraints are needed to fully define the problem. The control panel geometry is created as an NFF file along with the buttons. Each button is a single object. Areas of the control panel have texture mapped images applied which inform the user of the progress of the synthesis operation. These texture maps are changed as a result of the solution sequence and provide valuable information to the designer.

COMPARISON TO EXISTING SOFTWARE

SphereVR can be compared most directly to McCarthy's SPHINX program. Each program has its advantages and disadvantages. Locating the design positions on the sphere is easier and more intuitive using SphereVR. In SPHINX the user only sees one side of the sphere at a time and positions are moved on the sphere using the three mouse buttons to control movement in two orthogonal directions and one rotation. If a position is desired on the back of the sphere, the mouse mode must be changed and the sphere rotated before the position can be located. In SphereVR the operator can move or walk around to the other side of the sphere to place the next position.

The approach to specifying the pivot locations and the constraints is different in the two programs. The SPHINX program provides two methods for specifying the design objective: by specifying the type of mechanism desired or by picking the pivots. A useful two-dimensional color-contoured graph presents a guide map of all of the solutions that will produce the desired motion. The solutions are classified as to what type of mechanism is produced, such as crank-rocker, double crank, etc. This can be used to select a mechanism for a specific application. The method would involve placing the positions on the sphere then selecting a type of mechanism from the guide map. Based on the selection, the fixed and moving pivots would be specified.

The other method provides the circle axis cone and the center axis cone to be used to specify the location of the moving and fixed pivots. The circle axis cone locates the set of positions on the sphere that also lie in the moving body which are candidates for the location of the moving pivots. The center axis cone locates the possible positions of the fixed pivots on the sphere. In SPHINX, the cone locations are shown directly on the sphere.

SphereVR leaves the specification of the pivot locations and the constraints totally up to the designer. This gives the designer more freedom, but at a price. It is common for the software to fail to find a valid solution and very little help is given to guide the designer as to how to change the pivot locations to arrive at a valid solution. Future developments of the SphereVR software will incorporate a guide map and the cone solutions as aids to the designer.

In conclusion, specifying the design positions of the coupler link and generally getting a feel for the shape and motion of the resulting mechanism are the strengths of the SphereVR program. Guiding the designer in the design process is a strength of the SPHINX program. Future developments of the SphereVR will focus on providing the designer with more useful information.

FUTURE WORK

There are several areas open to future work. The Newton-Raphson method did not always converge for a given set of initial design parameters and constraints. A more robust method of synthesis could be incorporated into this program without undermining the virtual environment already created. In addition, this software could be expanded to synthesize more general spatial mechanisms.

Some indication of whether a valid solution exists would be useful. There are several mathematical tests that could be computed to give more information to the designer at an early stage to steer the design parameter specification into areas of feasible designs. Methods implemented by Chuan, Strong, and Waldron [14] and others could be evaluated and perhaps incorporated into this program.

As mentioned earlier, providing the designer with more guidance would also be helpful. Plotting the circle point and center point curves on the surface of the sphere will show the designer areas that may yield valid solutions. The guide map approach can also be implemented. Further evaluation of the SphereVR software in real applications is also planned. In addition, obstacles will be placed in the virtual environment and collision detection will be implemented. Mechanisms are rarely designed to be used by themselves, but more often are needed in constrained areas where obstacles must be avoided.

In terms of the virtual reality software and hardware used for this project, one improvement would be to use a less directional 3D selection device. The Logitech mouse is restricted to motion in a plane parallel to the receiver face. This eliminates full three-dimensional motion of the user. Either the receiver could be suspended from the ceiling and the transmitter mounted on the top of the hand or another type of position sensor could be incorporated. In addition, research into the degree of immersion necessary to perform spatial mechanism design is needed.

CONCLUSION

This paper has described the development and operation of a virtual environment for the synthesis of a special class of spatial mechanisms: spherical four-bar mechanisms. A brief history of graphical interface development for mechanism design was followed by a description of the operation of the virtual reality program developed by the authors. Challenges addressed during the development of the virtual environment were also presented.

The real power of VR comes into play when operating in two areas of this program: specification of the initial design constraints and validation of the resultant design. The designer is able to use natural hand motions to "pick and place" design parameters and natural head motions to look at the constraint surface and initial parameters before the solution is attempted. The control panel is a constant message board that informs the designer what is required next in the solution process. After the mechanism has been synthesized, the user can move into the sphere, walk around the sphere, move one of the pivot links, and/or watch the animation of the mechanism from any angle in order to get a feel for the mechanism that has been designed. The virtual environment presents a fully three-dimensional space in which to validate the design. Redesign is easily accomplished by changing constraints, moving pivots or moving the arrowheads.

In conclusion, a virtual reality interface was developed to aid in the synthesis of spherical four-bar mechanisms. Several challenges inherent in spatial mechanism design were addressed in the virtual environment. We believe this software holds the promise of providing a fully three-dimensional environment in which to perform spatial mechanism design.

ACKNOWLEDGEMENTS

The authors wish to thank the Iowa Center for Emerging Manufacturing Technology at Iowa State University for the support for this project.

REFERENCES

1. Freudenstein, F. , "Approximate Synthesis of Four-Bar Linkages", *Trans. ASME*, vol. 77, 1955, pp. 853-861.

2. Sheth, P.N. and Uicker, J.J., "IMP (Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkages," *Trans. ASME, Journal of Engineering for Industry*, May 1972, pp. 454 - 464.

3. Kaufman, R.E., "Mechanism Design by Computer," *Machine Design*, pp. 94-100, 1978.

4. Erdman, A.G., and Gustafson, J.E., "LINCAGES: Linkages INteractive Computer Analysis and Graphically Enhanced Synthesis Package", ASME paper 77-DET-5, presented at the ASME Design Engineering Technical Conference, Sept. 26 - 30, 1977.

5. Waldron, K.J., and Song, S. M. "Theoretical and Numerical Improvements to an Interactive Linkage Design Program, RECSYN," *Proceedings of the Seventh Applied Mechanisms Conference*, Kansas City, MO, December, 1981.

6. Thatch, B.R. and Myklebust, A., "A PHIGS-Based Graphics Input Interface for Spatial-Mechanism Design," *IEEE Computer Graphics and Applications*, vol. 8, March 1988, pp. 26 - 38.

7. Chen, X.Q. and Erdman, A.G., "Systematic Synthesis of Spherical Four-Bar Mechanisms," First National Applied Mechanisms and Robotics Conference, University of Cincinnati, Nov. 1989, paper No. 89AMR-78-5.

8. Larochelle, P., *Sphinx: Software for Synthesizing Spherical 4R Mechanisms: User's Guide*, Department of Mechanical Engineering, University of California, Irvine, CA, 1993.

9. Thoreson, J. and Erdman, A.G., "Designing Mechanisms for Production Equipment," *Machine Design*, vol. 60, October 6, 1988, pp. 113 - 117.

10. Barris, W.C., Kota, S., Riley, D.R. and Erdman, A.G., "Mechanism Synthesis Using the Workstation Environment," *IEEE Computer Graphics and Applications*, March 1988, pp. 39-50.

11. Aukstakalnis, S. and Blatner, D., *The Art and Science of Virtual Reality: Silicon Mirage*, Peachpit Press, Berkeley, CA, 1992.

12. Sandor, G.N. and Erdman, A.G., *Advanced Mechanism Design: Analysis and Synthesis*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.

13. Suh, C. H. and Radcliffe, C. W. , "Synthesis of Spherical Linkages With Use of the Displacement Matrix." *Journal of Engineering for Industry, Trans.* ASME, Series B, vol. 89, 1967, pp. 215-221.

14. Chuang, J.C. , Strong, R.T., and Waldron, K.J., "Implementation of Solution Rectification Techniques in an Interactive Linkage Synthesis Program," *Trans. ASME, Journal of Mechanical Design*, vol. 103, July 1981, pp. 657 - 664.