**T. J. Furlong**
Systems Engineer
Creative Optics, Inc.
Bedford, NH 03110
tfurlong@creative-optics.com

**J. M. Vance**
Associate Professor
Department of Mechanical Engineering
Virtual Reality Applications Center
Iowa State University
Ames, IA 50011
jmvance@iastate.edu

**P. M. Larochelle**
Assistant Professor
Department of Mechanical Engineering
Florida Institute of Technology
Melbourne, FL 32901-6975
pierre@gretzky.me.fit.edu

# Spherical Mechanism Synthesis in Virtual Reality

*This paper presents a new approach to using virtual reality (VR) to design spherical mechanisms. VR provides a three-dimensional (3-D) design space where a designer can input design positions using a combination of hand gestures and motions and view the resultant mechanism in stereo using natural head movement to change the viewpoint. Because of the three-dimensional nature of the design and verification of spherical mechanisms, VR is examined as a new design interface in this research. In addition to providing a VR environment for design, the research presented in this paper has focused on developing a "design in context" approach to spherical mechanism design. Previous design methods have involved placing coordinate frames along the surface of a constraint sphere. The new "design in context" approach allows a designer to freely place geometric models of movable objects inside an environment consisting of fixed objects. The fixed objects could either act as a base for a mechanism or be potential sources of interference with the motion of the mechanism. This approach allows a designer to perform kinematic synthesis of a mechanism while giving consideration to the interaction of that mechanism with its application environment.*

## Introduction

Erdman and Sandor (1991) define a mechanism as "a mechanical device that has the purpose of transferring motion and/or force from a source to an output." Mechanisms are generally composed of rigid links connected by joints. Typically, one or more links are grounded to a fixed or moving reference frame. Mechanical engineers are often required to design mechanisms to perform tasks separately or as part of a larger machine.

In 1959, Freudenstein was the first to use a computer to design mechanisms (Freudenstein and Sandor, 1959). Since then, a number of computer aided mechanism design packages have been developed, including KINSYN, LINCAGES, and RECSYN (Kaufman, 1978; Erdman and Gustafson, 1977; Waldron and Song, 1981).

Traditionally, mechanism design has concentrated on synthesis of planar motion, but in the late 1980's people began investigating the use of computers for spatial mechanism design. Thatch and Myklebust (1988) recognized that one of the difficulties in developing computer-aided spatial mechanism design software was specifying the three-dimensional input parameters, so they developed a package called Mechin to assist designers in specifying these input parameters. In 1989, LINCAGES was enhanced to allow synthesis of spherical four-bars (Chen and Erdman, 1989). In 1993, McCarthy and Larochelle introduced Sphinx, which was written expressly for synthesis of spherical four-bar linkages (Larochelle et al., 1993). More recently, SphinxPC has brought spherical mechanism design to the Windows platform (Ruth and McCarthy, 1997).

It can be difficult to make design choices in spherical mechanism synthesis because the designer must visualize and interact in three dimensions. Planar mechanism synthesis requires only two-dimensional (2-D) display and interaction, so it is well suited to the current human-computer interface (HCI) of mouse and monitor. Sphinx and SphinxPC require users to interact with 3-D objects, but they still use the traditional 2-D HCI. Osborn and Vance (1995) recognized spherical mechanism synthesis as a potential application of virtual reality technology. It was believed that the 3-D visualization and interaction offered by VR could assist a designer by adding a necessary dimension to the HCI.

Contributed by the Design Automation Committee for publication in the Journal of Mechanical Design. Manuscript received Oct. 1998; revised Sept. 1999. Associate Technical Editor: A. Diaz.

Isis, the program described in this paper, is the third generation of VR spherical mechanism design software developed at Iowa State University. The first program, SphereVR, had a user place coordinate frames on a "great sphere". This program used a Newton-Raphson iterative approach to solving the non-linear equations which resulted from a dyad approach to mechanism synthesis (Osborn and Vance, 1995). This initial exploration of VR for spherical mechanism design was followed by a second program called VEMECS (Virtual Environment for MEChanism Synthesis) (Kraal and Vance, 1999). Collaboration with the authors of Sphinx provided stronger solution algorithms, and VEMECS essentially became a VR interface to the Sphinx software. A study comparing the visualization and interaction methods of VEMECS and Sphinx warranted further research into the interface of VR spherical mechanism design programs (Evans, et al., 1999). The topic of this paper is the third generation of VR spherical mechanism synthesis, which is called Isis. Isis uses the latest versions of the Sphinx synthesis and analysis computation routines developed at Florida Institute of Technology, but it employs new forms of interaction and a "design in context" approach that involves working with geometrical models instead of coordinate frames during synthesis.

## Spherical Four-Bar Linkages

A spherical four-bar linkage consists of four links connected by four revolute, or pin, joints. In contrast to planar four-bar linkages, the revolute joint axes of a spherical four-bar converge at a point, and the output motion traces out a path along the surface of a sphere. A spherical mechanism is depicted in Fig. 1.

Spherical mechanisms allow an engineer to create a linkage to perform spatial motion. Prototypes have been developed to replace coupled planar linkages in certain applications by allowing smooth spatial motion with fewer moving parts. Spherical mechanisms have been proposed for use in devices to assist the handicapped, and it is possible that they could replace robots in performing repetitive tasks such as those required in manufacturing processes.

## Kinematic Synthesis of Spherical Four-Bars

The computer-aided design packages for spherical mechanisms mentioned above are dedicated to synthesizing spherical mechanisms for moving a body through a sequence of prescribed orientations in space. This task is referred to as "rigid-body guidance" by Suh and Radcliffe (1978) and as "motion generation" by

  

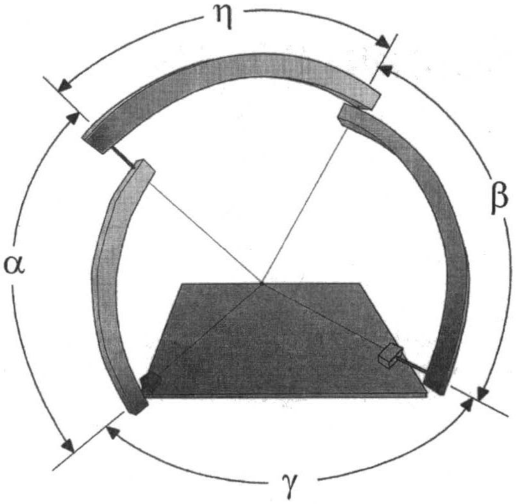


Fig. 1 Spherical four-bar mechanism

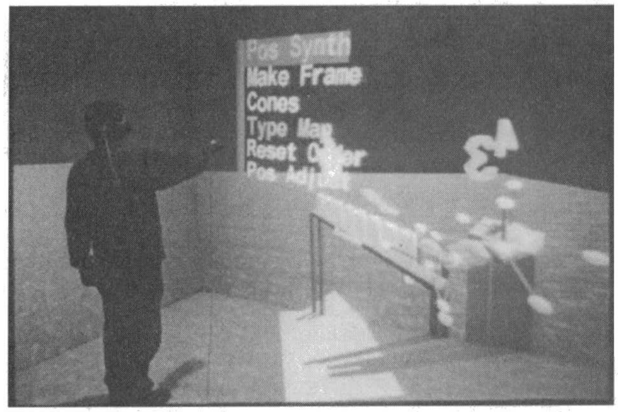


Fig. 2 Isis displayed in the C2 virtual environment

Erdman and Sandor (1977). Sphinx, SphinxPC, VEMECS, and Isis are dedicated to synthesizing mechanisms which guide a body through four orientations.

Once the four orientations are prescribed, the spherical generalization of Burmester's planar theory is employed to determine the set of all mechanisms which accomplish the task (Larochelle et al., 1993). The result of Burmester's solution to four orientations are two cubic cones referred to as the fixed axis cone and the moving axis cone. The fixed axis cone is the set of all fixed axes of spherical RR dyads that will guide the moving body through the four prescribed orientations. The moving axis cone is the set of corresponding moving axes of the spherical RR dyads. A spherical four-bar mechanism may be viewed as an assemblage of two spherical RR dyads, where a dyad consists of a fixed and moving axis pair. Hence, we have a two-dimensional solution set, and selecting two points from either of the cones defines the two dyads which are then assembled to define a spherical four-bar mechanism that accomplishes the task.

The fixed and moving axis cones define all of the mechanisms which will guide a body through the four orientations. However, Burmester's solution is not sufficient to arrive at a practical solution. Further analyses must be performed to examine the type of mechanism, whether or not the mechanism is input drivable, and if the four orientations are reached in the desired order. These performance criteria are collectively referred to as solution rectification by Waldron and Strong (1978). Sphinx, SphinxPC, and Isis perform the necessary solution rectification and present the results as a "type map" (Murray and McCarthy, 1995). The type map displays all of the solutions generated by Burmester's theory color-coded by mechanism type. Moreover, the type map is equipped with a filter to display only those mechanisms that are input drivable and reach the orientations in the desired order. The result is a tool which enables the designer to select a mechanism which is a practical solution to the prescribed task.

## The Isis Environment

*Peripherals.* Isis can be used in a head-mounted display (HMD), on a projection screen using CrystalEyes stereo shutter glasses, or in Iowa State University's C2, a CAVE™-like surround-screen virtual reality room. Fakespace PINCH Gloves may be used for interaction in conjunction with any of the display devices mentioned above. PINCH Gloves register contact between a user's fingers, allowing gestural input to the program. Ascension Flock of Birds magnetic trackers are used to track the motion of a user's head and hand. Figure 2 shows a user interacting with the Isis software in the C2 virtual reality environment. The user can also interact with the program using a standard mouse and monitor when VR peripherals are not available.

Because of the 3-D nature of specifying desired positions and of verifying the resultant mechanism design, the immersive nature of the C2 and the HMD greatly enhance the design experience of the user of Isis. It was our experience that projecting Isis on a one wall screen or interacting with Isis using a standard mouse and monitor was insufficient to provide the truly immersive virtual environment which is needed for spherical mechanism design. So, although the program can be used with many different interfaces, the C2 or the HMD is preferred.

*Interaction.* Three basic PINCH Glove gestures are used in the program, and the meaning of each gesture is kept consistent throughout. One gesture is used for grasping an object and moving it within the space. A second gesture selects menu items and is used for interactions that fall outside of the standard grasp and move mode. The third basic gesture is used to increment through the steps that must be taken to synthesize a mechanism. Each gesture requires the user to simply touch one finger to the thumb.

To provide higher level functionality to Isis, a 3-D menu system was created. Menus are used for file manipulation and for choosing options that would be inconvenient to assign to a single gesture. The menus are virtual objects consisting of text items and a menu bar. The menus may be grasped by the user's virtual hand and repositioned in space. Pointing at a menu item and gesturing appropriately selects that item. The program's main menu is depicted in Fig. 3.

*Design Methodology.* The Isis environment begins as an empty space. This is a departure from previous spherical mechanism design programs, which begin with a large sphere as the only

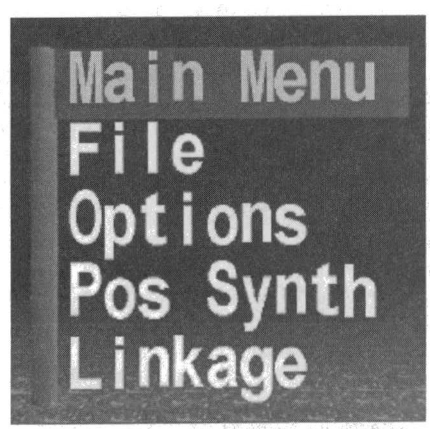


Fig. 3 Isis main menu

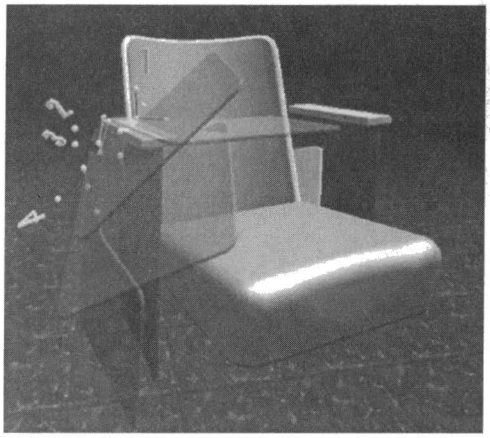

Fig. 4 Chair base and tray instances

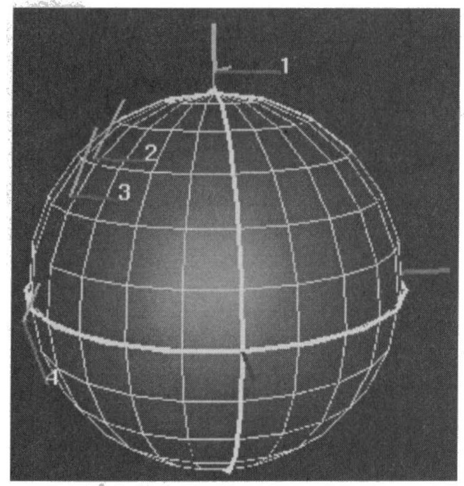


Fig. 5 Tray positions in sphinx

object in the design environment. The goal of the initial empty space is to provide the user with a completely customizable design environment that is not biased by the presence of a design sphere. In Isis, a sphere is only introduced into the environment when it is necessary to show the spherical constraint that a user has created by the placement of design positions. In Isis, emphasis is placed on the spatial task, not on interacting with a design sphere which has no physical counterpart in the final design.

When designing a mechanism, the user has the option of working with geometric models of actual parts. For example, the lecture hall chair in Fig. 4 was loaded in as the base model for the mechanism design. The user may also use a geometry file to define a position synthesis task. The tray tables in Fig. 4 are movable instances of a model brought into Isis for this purpose. Geometry files can come from CAD packages such as AutoCAD and Pro/Engineer, and from modeling packages such as MultiGen and 3D Studio.

Once a movable geometry has been loaded, the user may grasp it and freely place it in space. To make it easier to place the geometry precisely, the user can turn on the option to constrain movement to either the X-Y, X-Z, or Y-Z plane of the global coordinate frame. To reduce visual clutter, the geometry defining a position synthesis task is rendered as semi-transparent. An instance of the geometry becomes opaque when it is manipulated, allowing the user to concentrate on that particular position.

Once the user has placed one position of the movable geometry, a second instance is created and moved to another position in space. Usually the first and second positions signify the desired beginning and ending positions for the motion of the linkage. The first position is placed freely in space. To guarantee purely spherical motion between the first two positions, the second position is constrained such that the z-axes of the first two moving frames intersect. Once the first two positions have been placed, the spherical constraint surface is defined and displayed. The remaining two positions are confined to the sphere defined by the first two position choices.

After four positions are placed, the user selects the desired order in which the mechanism should move through the positions. At this point, the fixed and moving axis cones or the type map may be generated. Isis gives the position information to the Sphinx computation routines, and the Sphinx algorithms return the appropriate solution information.

The ability to work with geometric models is a feature that sets Isis apart from previous approaches to design. Motion synthesis within the context of its application allows users to see right away whether or not a mechanism is feasible for its intended use. Unwanted collisions between objects can instantly be seen, and the design can be altered immediately. Figure 4 depicts four instances of a tray model placed around a chair that will act as a base for a mechanism. It can clearly be seen that the intended task of the mechanism in this example will be to move the tray from its working position to its stowed location. This would not be evident if a user were to work exclusively with coordinate frames, as can be seen in Fig. 5, which shows the same four positions in Sphinx. The design emphasis in Isis is placed on defining the task of moving the tray table, as opposed to placing coordinate frames on a constraint sphere.

Figure 6 shows another example of a spherical motion synthesis task that benefits from using geometric models. Here, a designer creates a mechanism that ensures that the soda can will travel along a path that is free of obstruction.

Figure 7 shows the prototype mechanism which was constructed based on the design created in the virtual environment for the soda can task.

*Cones.* Sphinx routines calculate the fixed and moving axis cones, and Isis displays these as red and blue 3-D curves. Users of Isis are able to see links appear as selections are made on the cones. When two links are chosen, a coupler curve appears, and a message box opens to tell the user what type of linkage has been synthesized. The user may then move the axes to see how the linkage and coupler curve change. Figure 8 shows a user creating a link by picking on the red fixed axis cone.

*Type Map.* As stated previously, a type map is a 2-D plot that displays the solutions generated by Burmester's theory which are color-coded by mechanism type. Isis takes advantage of the VR display and brings the type map into three dimensions. Linkages

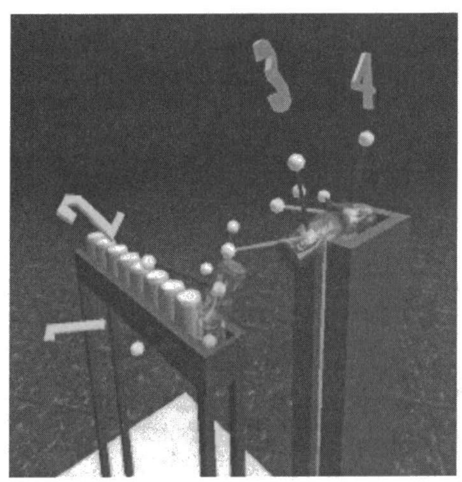


Fig. 6 Soda can task

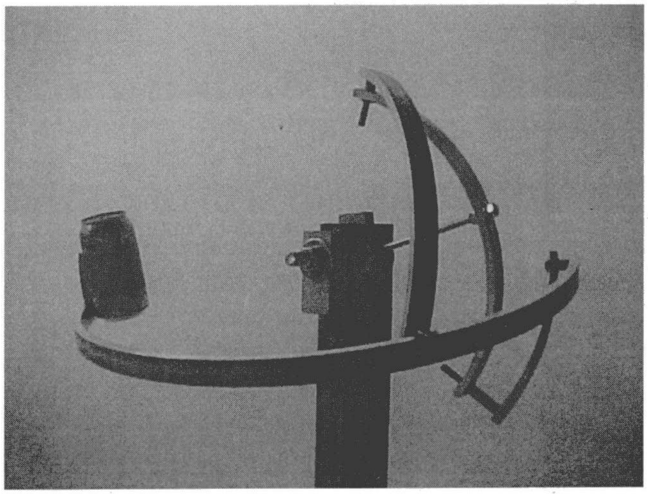

Fig. 7 Actual soda can mechanism

that pass the test of input drivability and reach the orientations in the desired order are shown on a plane above the whole solution set, and those solutions that do not pass the tests are darkened on the type map. The result of this is a display that shows good and bad mechanisms together, with the good mechanisms easily distinguishable from the undesirable ones.

A user selects a point in a colored region based on the type of linkage that he or she is trying to synthesize, and the linkage corresponding to that point appears along with information about the linkage type and folding condition. In the example shown in Fig. 9, one can see the type map, which displays the better mechanisms as areas of bright blue and yellow. Additionally, a complete mechanism resulting from the type map choice is shown with its coupler curve depicted in yellow. To the right of the mechanism is a message box containing confirmation of the mechanism type that has been chosen and information about the folding condition of the linkage.

*Design Verification.* Once a mechanism has been synthesized in Isis, it may be animated to verify that it completes the required task. The transparent positions remain present, and an opaque instance of the moving geometry moves along with the coupler link of the mechanism. Designers can observe how smoothly a mechanism moves, and they can see whether or not objects will collide during motion of the mechanism. Because of the availability of head tracking in the virtual reality environment, designers are allowed to move around the design and investigate the motion

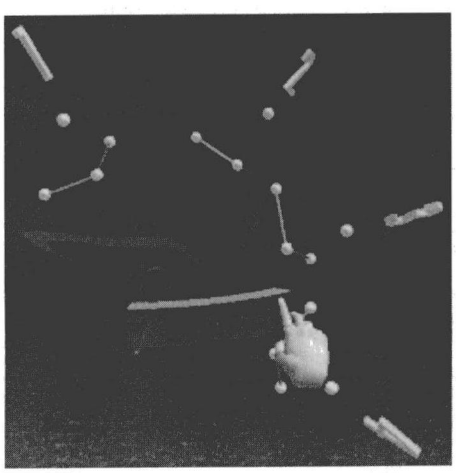

Fig. 8 Picking cones

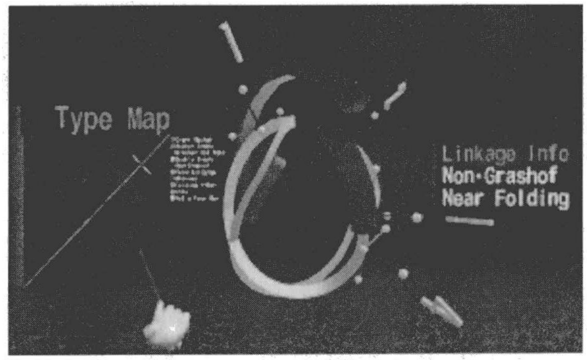


Fig. 9 Type map interaction

of the mechanism from many different angles. It is very intuitive to investigate the operation of the mechanism because the designer's viewpoint changes according to natural human motions.

*File Manipulation.* After a user has synthesized a mechanism, an output file can be generated. The file written by Isis is an augmented Sphinx file, which is readable by Sphinx. The augmented file contains information such as sphere radius, sphere position, geometric model filenames, and geometric model positions. This information is used to reconstruct the "design-in-context" environment when a file is loaded into Isis. Standard (non-augmented) Sphinx files can also be loaded into Isis. When loading a standard Sphinx file, Isis will use a default sphere radius and position, and it will not attempt to load any geometric models.

*Summary of Program Usage.* Figure 10 shows a flowchart summarizing the procedure taken to synthesize a spherical mechanism in Isis. Parallel portions of the flowchart indicate optional paths. For example, a user may bring in geometry or simply use coordinate frames to define a synthesis task. Alternatively, a mechanism previously saved from Sphinx or Isis may be loaded into the program. To generate or redesign a mechanism, either cones or a type map may be used. Users may alter positions and regenerate the cones or type map until a good solution is obtained.

In Fig. 10, the gray boxes indicate parts of the program that utilize Sphinx functions. Some of the Sphinx functions have remained untouched, but others have been modified to fill Isis data structures and create geometry for display by WorldToolKit.

## Conclusions

The presence of virtual reality in engineering design is growing, and the work presented here is a reflection of that. The visualization benefits of VR have long been known, and the modes of interaction in VR are converging towards a naturalistic interface. Isis is written for use with the latest VR devices, and interaction within the program is not an adaptation of a workstation interface. It is written expressly to be a virtual environment in which interaction and viewing are intuitive. The 3-D display of a mechanism, the availability of information about the mechanism, and the "design in context" methodology present a complete picture of a design, so there are fewer surprises when a mechanism is brought from the virtual world to the real world.

## Future Work

New methods of task specification are being developed at Florida Tech. The research is focusing on novel methods for prescribing the desired positions of a moving body. These new methods are intended to enhance the "design in context" capabilities of Isis. One method allows a user to place any number of positions freely in space and then the optimal design sphere is automatically determined as well as the corresponding positions on the sphere. Another approach allows the user to specify one position of the

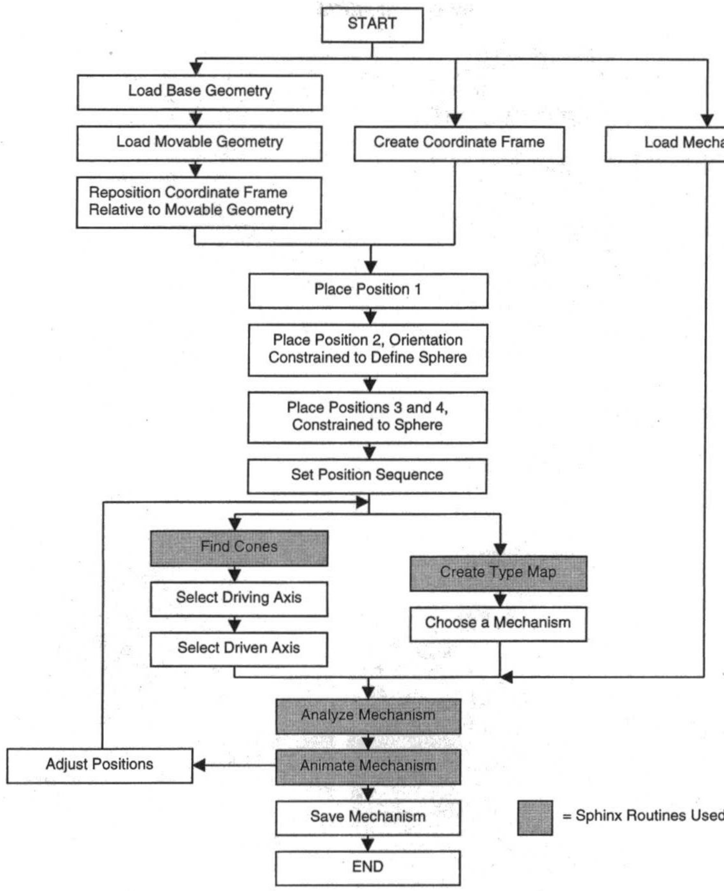


**Fig. 10 Diagram of program usage**

moving body which is to be realized exactly and a set of positions which serve to guide or shape the motion as desired. Again, the optimal design sphere and the corresponding positions are automatically computed. The goal of this research is to allow the designer to specify the task in the physical workspace without imposing the artificial constraints associated with a design sphere. Incorporation of this research into Isis is a planned future improvement.

Another feature planned for Isis is automatic collision detection between fixed and movable geometry. This could be done by building upon the swept volume research of Ling and Hu (1997) or by using a collision detection library such as RAPID (Gottschalk et al., 1996). Presently the user has to visually inspect the mechanism to check for collisions, but checking collisions with the computer and restricting the motion of the mechanism appropriately would relieve the designer of some work.

While interaction in Isis is fairly intuitive, placing the intermediate positions in their desired orientation while constrained to the design sphere needs improvement. The difficulty lies in the fact that although the visual representation of the movable geometry is constrained to the spherical design surface, the user's hand is not constrained to any surface and is free to move in the 3-D design space. Therefore, in this implementation of Isis, precise positioning of the movable geometry is still not natural and the manipulation of the geometry in these instances is not yet intuitive. Other methods of interactively placing these positions while constrained to the design sphere need to be investigated.

The real proof of the effectiveness of an environment such as Isis will be using it to design practical mechanisms. Material handling processes are being examined to see if an Isis-designed mechanism can be effective in such an environment. Other uses for spherical four-bar mechanisms are being sought as well. Applying Isis to practical synthesis tasks will enable designers to evaluate the effectiveness of the VR implementation of our "design in context" approach to spherical mechanism design.

## References


Chen, X. Q., and Erdman, A. G., "Systematic Synthesis of Spherical Four-Bar Mechanisms," *First National Applied Mechanisms and Robotics Conference*, University of Cincinnati, 89AMR-78-5, Nov. 1989.

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

Erdman, A., and Sandor, G. N., *Mechanism Design: Analysis and Synthesis*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

Erdman, A., and Sandor, G. N., *Advanced Mechanism Design: Analysis and Synthesis*, Prentice Hall, Englewood Cliffs, New Jersey, 1997.

Evans, P. T., Vance, J. M., and Dark, V. J., "Assessing the Effectiveness of Traditional and Virtual Reality Interfaces in Spherical Mechanism Design," ASME Journal of Mechanical Design, in press, 1999.

Freudenstein, F., and Sandor, G. N., "Synthesis of Path-Generating Mechanisms by Means of a Programmed Digital Computer," *ASME Journal of Engineering for Industry* Vol. 81B, No. 2 pp. 159–168, May 1959.

Gottschalk, S., Lin, M. C., and Manocha, D., "OBB-Tree: A Hierarchical Structure for Rapid Interference Detection," *Proceedings of ACM SIGGRAPH '96*, pp. 171–180, New Orleans, LA, 1996.

Kaufman, R. E., "Mechanism Design by Computer," *Machine Design*, Vol. 50, No. 24 pp. 94–100, Oct. 26, 1978.

Kraal, J. C., and Vance, J. M., "VEMECS: A Virtual Reality Interface for Spherical Mechanism Design," submitted to *The Journal of Engineering Design*, 1999.

Larochelle, P. M., Dooley, J. R., Murray, A. P., and McCarthy, J. M., "SPHINX: Software for Synthesizing Spherical 4R Mechanisms," *NSF Design and Manufacturing Systems Conference*, 1, pp. 607–611, Jan. 1993.

Ling, Z., and Hu, Z., "Use of Swept Volumes in the Design of Interference Free Spatial Mechanisms," *Mechanism and Machine Theory*, Vol. 32, No. 4, pp. 459–476, 1997.

McCarthy, J. M., "The Synthesis of Planar RR and Spatial CC Chains and the Equation of a Triangle," ASME Journal of Mechanical Design, Vol. 117(B), pp. 101–106, June 1995.

Murray, A., and McCarthy, J., "A Linkage Type Map for Spherical 4 Position Synthesis," *Proceedings of the 1995 ASME Design Technical Conferences*, Boston, MA, DE-82:833–838, Sept. 1995.

Osborn, S. W., and Vance, J. M., "A Virtual Environment for Synthesizing

Spherical Four-Bar Mechanisms," *Proceedings of the 1995 Design Engineering Technical Conferences,* Boston, MA, DE-83:885–892, Sept. 1995.

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

Suh, C. H., and Radcliffe, C. W., *Kinematics and Mechanism Design,* John Wiley & Sons, New York, 1978.

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

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, Dec. 1981.

Waldron, K. J., and Strong, R. T., "Improved Solutions of the Branch and Order Problems of Burmester Linkage Synthesis," *Mechanism and Machine Theory,* Vol. 13, No. 2 pp. 199–207, 1978.