

6-2009

Collaborative Research: Constraint-based Compliant Mechanism Design Using Virtual Reality as a Design Interface

Judy M. Vance

Iowa State University, jmvance@iastate.edu

James H. Oliver

Iowa State University, oliver@iastate.edu

Utkarsh Seth

Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/me_conf



Part of the [Computer-Aided Engineering and Design Commons](#), and the [Graphics and Human Computer Interfaces Commons](#)

Recommended Citation

Vance, Judy M.; Oliver, James H.; and Seth, Utkarsh, "Collaborative Research: Constraint-based Compliant Mechanism Design Using Virtual Reality as a Design Interface" (2009). *Mechanical Engineering Conference Presentations, Papers, and Proceedings*. 116.
http://lib.dr.iastate.edu/me_conf/116

This Conference Proceeding is brought to you for free and open access by the Mechanical Engineering at Iowa State University Digital Repository. It has been accepted for inclusion in Mechanical Engineering Conference Presentations, Papers, and Proceedings by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Collaborative Research: Constraint-based Compliant Mechanism Design Using Virtual Reality as a Design Interface

Abstract

The objective of this research is to develop an immersive interface and a design algorithm to facilitate the synthesis of compliant mechanisms from a user-centered design perspective. Compliant mechanisms are mechanical devices which produce motion or force through deflection or flexibility of their parts. Using the constraint-based method of design, the design process relies on the designer to identify the appropriate constraint sets to match the desired motion. Currently this ability requires considerable prior knowledge of how non-linear flexible members produce motion. As a result, the design process is based primarily on the designer's previous experience and intuition. A user centered methodology towards the design of compliant mechanisms is suggested where the interface guides the designer throughout the design process. This research proposes an algorithm which places an abstract layer between the designer and the design process thereby hiding the complex mathematical calculations and providing an immersive virtual environment for user interaction. A virtual reality (VR) immersive interface lets the user interact with the problem at hand in a natural way with hand gestures, head motion, etc. This enables the designer to input the intended motion path by simply grabbing and moving the object and letting the system decide which constraint spaces apply. The user-centered paradigm supports an approach that focuses on the designer defining the motion and the system generating the constraint sets, instead of the current method which relies heavily on the designer's intuition to place constraints. The input from the user drives the design process and the system produces a set of possible solutions. This research results in an intelligent design framework that will allow a broader group of engineers to design complex compliant mechanisms, giving them new options to draw upon when searching for design solutions to critical problems

Keywords

Virtual Reality Applications Center

Disciplines

Computer-Aided Engineering and Design | Graphics and Human Computer Interfaces

Comments

This is a conference proceeding from *2009 NSF Engineering Research and Innovation Conference Proceedings* (2009). Posted with permission.

NSF GRANT#: CMMI-0457041

NSF PROGRAM NAME: Engineering Design and Innovation

Collaborative Research: Constraint-based Compliant Mechanism Design Using Virtual Reality as a Design Interface

Dr. Judy M. Vance

Department of Mechanical Engineering
Virtual Reality Application Center
Iowa State University
Ames, IA 50011
jmvance@iastate.edu

Dr. James H. Oliver

Department of Mechanical Engineering
Virtual Reality Application Center
Iowa State University
Ames, IA 50011
oliver@iastate.edu

Utkarsh Seth

Virtual Reality Application Center
Iowa State University
Ames, IA 50011
useth@iastate.edu

ABSTRACT

The objective of this research is to develop an immersive interface and a design algorithm to facilitate the synthesis of compliant mechanisms from a user-centered design perspective. Compliant mechanisms are mechanical devices which produce motion or force through deflection or flexibility of their parts. Using the constraint-based method of design, the design process relies on the designer to identify the appropriate constraint sets to match the desired motion. Currently this ability requires considerable prior knowledge of how non-linear flexible members produce motion. As a result, the design process is based primarily on the designer's previous experience and intuition.

A user centered methodology towards the design of compliant mechanisms is suggested where the interface guides the designer throughout the design process. This research proposes an algorithm which places an abstract layer between the designer and the design process thereby hiding the complex mathematical calculations and providing an immersive virtual environment for user interaction. A virtual reality (VR) immersive interface lets the user interact with the problem at hand in a natural way with hand gestures, head motion, etc. This enables the designer to input the intended motion path by simply grabbing and moving the object and letting the system decide which constraint spaces apply. The user-centered paradigm supports an approach that focuses on the designer defining the motion and the system generating the constraint sets, instead of the current method which relies heavily on the designer's intuition to place constraints. The

input from the user drives the design process and the system produces a set of possible solutions.

This research results in an intelligent design framework that will allow a broader group of engineers to design complex compliant mechanisms, giving them new options to draw upon when searching for design solutions to critical problems

1 Introduction

The largest challenge in designing a compliant mechanism [1] is the difficulty in understanding the motion of the compliant members. The deflection of the compliant members is complex due to the geometric non-linearity present in the members. Due to this, the design of such mechanisms has been dependent on the experience and intuitiveness of the designer. This has prevented novice designers to enter the domain and apply their skills to the same.

In the mechanism design field, significant research has been performed on applying computational techniques for the synthesis of compliant mechanisms to achieve a defined motion. The most often used approaches in the area are the pseudo rigid body model approach [2] and topological synthesis [3-8]. In the pseudo rigid body model, a rigid body analysis method is used in the analysis of compliant mechanisms. This approach models a compliant mechanism as a rigid body which allows the use of rigid body theories and methodologies [9-10]. Validation and verification of the results are important because of the simplifications inherent in this model of the system. The topological synthesis method

relies on optimization methods to arrive at an optimum structural topology to achieve specified motion requirements. It models the mechanisms as a series of several link members of different sizes which together perform the desired motion. In both of the above methods prior experience and mechanism design knowledge is needed for successful completion of the design. The third approach, on which this research is based, is the constraint based design approach. In this approach, introduced by Maxwell [11], the position and orientation of constraints applied to a body at any given instance defines its motion. Our approach comes in useful as it helps the designer in visualizing motions and ultimately designing the desired mechanism. We also propose a user centered methodology for the design of compliant mechanisms in a virtual reality environment.

2. Background

2.1 FACT method

The freedom space or freedom topology represents the object's allowable motion in space. The constraint space represents the restricted motions in space. Researchers at the Precision Engineering Lab at MIT [12] have extended Blanding's theory to produce a series of geometric representations for freedom and constraint spaces in terms of allowable motions of the body. The method they developed is known as FACT (Freedom and Constraint Based Topologies). FACT deals with the different constraint and freedom spaces by dividing them in different CASEs and TYPEs. It consists of a catalogue of all the possible freedom and constraint space sets which could apply for a given motion. The CASE in the FACT method defines the number of constraints applied on the body. For example, CASE # 1 denotes mechanisms with one constraint which results in five degrees of freedom as only one constraint is applied. The TYPEs within a CASE defines ways in which degrees of freedom of a body could be achieved. Therefore, there are several TYPEs in each CASE.

For every constraint space produced, there is a specific freedom space. FACT provides geometric representations of constraint and freedom sets of all the CASEs and TYPEs. Though the method has all the representations for freedom and constraint spaces, it still requires considerable effort on the part of the designer to understand them before their application to a given motion.

2.2 Screw Theory

In a rigid body motion, a general motion could be described using a screw. Geometrically a screw could be represented as a rotation about a line in space and a translation about that line. This line is known as the screw axis. Mathematically a screw motion is described with a twist which is a six dimensional vector representing the linear and angular velocities of the body, written as

$$\hat{T} = (\boldsymbol{\Omega} \quad ; \quad \mathbf{V}) = (\omega \mathbf{s} \quad ; \quad \mathbf{c} \times \omega \mathbf{s} + v \mathbf{s}) = (\omega \mathbf{s} \quad ; \quad \mathbf{c} \times \omega \mathbf{s} + p \omega \mathbf{s})$$

where $\boldsymbol{\Omega}$ is the angular velocity, \mathbf{V} is the linear velocity, \mathbf{s} is the vector denoting the twist axis, \mathbf{c} is a point on the axis, ω is the magnitude of angular velocity along the axis, v is the partial linear velocity along the axis and p is the pitch defined as v/ω .

The constraint or restricted motion in space is represented by a *wrench* which consists of two vectors representing a force \mathbf{F} and a couple (moment) \mathbf{M} acting on a rigid body, written as,

$$\hat{W} = (\mathbf{F} \quad ; \quad \mathbf{M}) = (f \mathbf{u} \quad ; \quad \mathbf{r} \times f \mathbf{u} + m \mathbf{u}) = (f \mathbf{u} \quad ; \quad \mathbf{r} \times f \mathbf{u} + q f \mathbf{u})$$

where vectors \mathbf{u} and \mathbf{r} denote the direction of and a point on the wrench axis respectively, scalars f and m are magnitude of the force and partial moment along the axis, coupled by a pitch $q=m/f$.

These two concepts are often known as duality [13] in kinematics and statics. Screw theory has been applied to the constraint based compliant mechanism design approach. Ball [14] was the first to formulate screw theory in a systematic way. Hunt [15] and Phillips [16,17] later developed the geometrical and mathematical representation of screws and screw systems. They used the screw theory for the synthesis and analysis of mechanisms. Since then, screw theory has also been applied to topology synthesis [18]. Kim [19] studied the characterization of compliant building blocks by utilizing the concept of eigen-twists and eigen-wrenches based on screw theory. Su [20] et al. proposed a screw theory based approach for the conceptual design of compliant mechanisms. In this approach, a freedom space (all allowable motions) is defined by a twist matrix given by

$$\Pi_T = \begin{bmatrix} \hat{T}_1 \\ \hat{T}_2 \\ \vdots \\ \hat{T}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega}_1 & ; & \mathbf{V}_1 \\ \boldsymbol{\Omega}_2 & ; & \mathbf{V}_2 \\ \vdots & ; & \vdots \\ \boldsymbol{\Omega}_f & ; & \mathbf{V}_f \end{bmatrix}$$

where f is the dimension of the freedom space and \hat{T}_i are basis twists that span the freedom space. For example, the freedom space generated by a serial chain of two intersecting revolute joints could be represented by

$$\hat{T} = k_1 \hat{T}_1 + k_2 \hat{T}_2 = (k_1 \boldsymbol{\Omega}_1 + k_2 \boldsymbol{\Omega}_2 \quad ; \quad \mathbf{c} \times (k_1 \boldsymbol{\Omega}_1 + k_2 \boldsymbol{\Omega}_2))$$

where $\hat{T}_1 = (\boldsymbol{\Omega}_1 \quad ; \quad \mathbf{c} \times \boldsymbol{\Omega}_1)$ and $\hat{T}_2 = (\boldsymbol{\Omega}_2 \quad ; \quad \mathbf{c} \times \boldsymbol{\Omega}_2)$ are the joint axes and the coefficients k_1 and k_2 can be viewed as the angular speeds of the joints. And \mathbf{c} is the intersection point. Any motion in this space is a rotation around the axis in the direction $k_1 \boldsymbol{\Omega}_1 + k_2 \boldsymbol{\Omega}_2$ through the point \mathbf{c} .

3. Methodology

3.1 Overview

FACT theory, as explained, gives geometric representations of the freedom and constraint spaces. For any designer to use it, he/she needs to know the intricacies as to what the freedom and constraint spaces mean and how have they been developed, before he/she could go forward to choose one of them for a given motion. Screw theory, although gives a mathematical approach to the solution, it requires the user to solve a number of equations and do mathematical calculations before reaching a final solution.

A gap exists in the design process such that there exist two approaches to design compliant mechanism but neither of them follows a user centered approach. Both of them call for the designer's attention towards steps which are not directly related to the compliant mechanism design process. Our approach tries to bridge this gap by using both of the above methodologies and follow a user centered design paradigm to give the user a 3 dimensional immersive interface to design compliant mechanisms. The interface helps the user during the design process through its intuitive user interface. Our approach we believe would enable even the novice designers to enter the compliant mechanism design domain as it abstracts the complex mathematical calculations from screw theory and does not rely on the user's understanding of the complexities of the geometric representations that lie behind the FACT method.

Before the design process even starts, a catalogue of the twist vector representations of all the freedom spaces is pre-calculated and stored. The freedom spaces define the allowable motions for a body. They could be represented as twist vector representations as explained in the previous sections. Now in the design process, when the user defines a motion, the twist vector representations of those motions are calculated. At this point the system has the twist vector representations of the user motion and also of the freedom spaces. The algorithm runs the user motion representations against all the freedom spaces representations to check under which CASE and TYPE the user motion falls under. This way, at the end of the algorithm the system knows the freedom space the user motion falls into. The system then displays the corresponding constraint space in front of the user to let the user select the constraints and proceed with the process.

For the proof of concept, we have developed a catalogue for CASE 3 TYPEs 1, 4 and 5 freedom spaces. It is to be significantly noted that the freedom spaces, as explained by Hopkins, are not associated with any coordinate system. They show no information about their orientation and location in space. To come up with a twist vector, we need to put the freedom spaces in a coordinate system. The freedom spaces could be attached to the coordinate system in three different ways where they could lie in any of the x-y, y-z or z-x planes. Therefore, there would be 3 twist representations of each freedom space.

The following section explains freedom space TYPEs 1, 4 and 5 within CASE 3 with their diagrammatic representation. Analysis of individual freedom spaces is done and the possible motions they represent are calculated mathematically in a twist vector form. We will go through the 3 TYPEs within CASE 3 and demonstrate how the freedom spaces could be aligned in a coordinate system.

CASE 3 TYPE 1

The CASE number represents the number of constraints applied on the body. Here as the CASE number is 3, there are 3 constraints applied and as a result 3 degrees of freedom of the body are free. This CASE and TYPE is geometrically represented as shown below [12]. The freedom space is

represented by a hoop and a plane. The hoop represents translation along the direction perpendicular to the axis of the hoop and the plane represents rotation about axes aligned with either side of the plane.

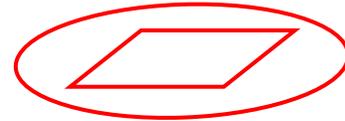


Figure 1: CASE 3 TYPE 1

We now associate this space to a coordinate system to arrive at the twist vector representation. In the twist vector representation, the top three elements represent rotation and bottom ones represent the translation components.

Table 1 : CASE 3 TYPE 1

| Freedom space in 3 orientations | Allowed motions | Twist representations |
|---------------------------------|-----------------|--|
| | | $\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ |
| | | $\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ |
| | | $\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ |

CASE 3 TYPE 4

CASE 3 TYPE 4 freedom space is represented as shown below. It is represented by a sphere with infinite freedom lines all passing through the center. This space represents rotation along the 3 axis and blocks any translation.

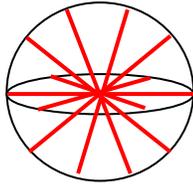


Figure 2: CASE 3 TYPE 4

Due to the symmetry of the freedom space, it is not relevant to display it in different orientations as they would all come out to be the same. Also, the twist vector of each representation would also come out to be the same.

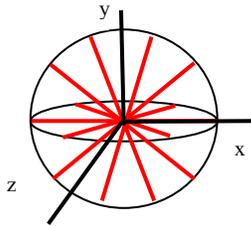


Figure 3: CASE 3 TYPE 4 with axis

The twist vector representation also would be the same for all the orientations given by

$$\text{Twist vector representation} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

CASE 3 TYPE 5

CASE 3 TYPE 5 is represented as shown below. It consists of a hoop and a pencil freedom space. The hoop represents translation and the pencil represents rotation motions.

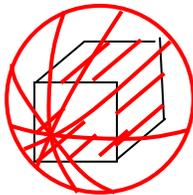


Figure 4 CASE 3 TYPE 5

Now we would associate the freedom space with a coordinate system and determine the twist representations.

We now have the twist representations of the desired freedom spaces (Table 2). Any user motion which falls into one of these freedom spaces would be recognized by the system and appropriate constraint spaces would appear.

Table 2: CASE 3 TYPE 4

| Freedom space in 3 orientations | Allowed motions | Twist representations |
|---------------------------------|-----------------|--|
| | | $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$ |
| | | $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$ |
| | | $\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$ |

4. Proposed solution

Now that the relationship between screw theory, compliant mechanism design and FACT has been developed, a tool can be developed for the design and analysis of compliant mechanisms. The solution proposed below follows a user-centered design paradigm where several interface design principles and theories have been followed to give an intuitive user interface to the user.

4.1 Scenario

A user gets in the immersive virtual reality environment and sees a virtual object in front. The user has a pre-defined goal for the desired motion path. User grabs the object and defines that path by rotating or translating the object. She/he marks every independent motion by explicitly telling the system about each one of them. This task is done by the use of a menu option which is selected once the user defines one motion. This way the system knows what independent motions the user has user defined and are thus stored in the system. Once the user is finished with defining the path, he/she selects the "Finish" option in the menu to let the system do the processing. The system then comes up with the appropriate constraint spaces, which allow the defined motion path. The user selects appropriate constraints from the space (guided by design principles) which then results in a physical mechanism.

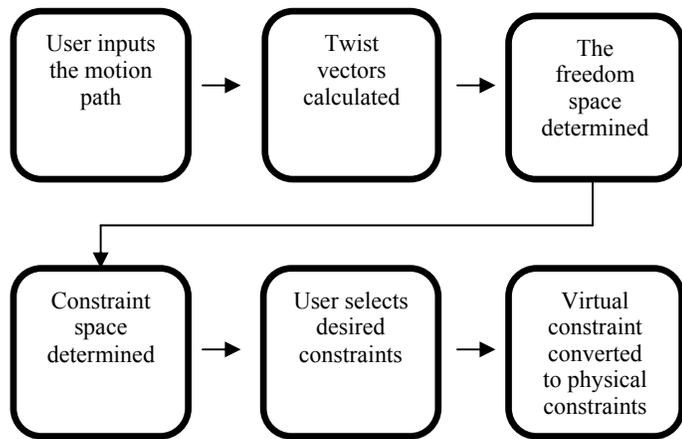


Figure 5 : Solution

4.1.1 STEP 1: User defines motion

The user grabs the object and locates it to a position by translating or rotating it. This defines the first motion. The user, as explained above, uses a menu option to declare the first independent motion. As the user does that, the object snaps back to the original position to let the user start from the beginning in case he/she wishes to define another motion. Once the user is finished, the system has $n + 1$ number of matrices ('n' number of positions & 1 starting position where $n \leq 6$).

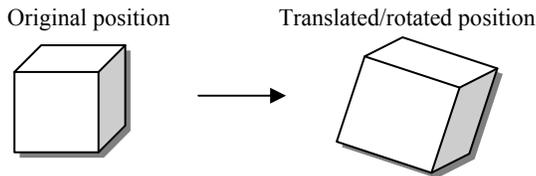


Figure 6: Object transformed to a new location

Once each independent motion is defined, the user might find it hard to locate it back to the exact original position to define the next motion. To assist the user, the system itself snaps the object back to the original position as soon as each motion is defined.

4.1.2 STEP 2: Twist vectors calculated

Once the system has the transformation matrices, they are converted to their twist vector representation using Screw theory.

4.1.3 STEP 3: Freedom space determined

After the completion of the second step, the system has the twist representations of the individual motions defined by the user. As explained above, the twist representations of the freedom spaces have already been pre calculated for comparison purposes. The next step involves determining the CASE and TYPE of the freedom space in which this user defined motion falls into. Once the system knows how many independent motions the user wants, the CASE number is automatically known. Once the CASE number is known, the next step is to determine the TYPE within that CASE. There could be 2 ways in which the user motions could be matched

to an appropriate freedom space. These ways depend upon the coordinate axis in which the user motions are defined.

1. □ If the user motions are defined along orthogonal axes, then in order to identify a freedom space a simple twist vector equivalency (user motion twist vector with the freedom space twist vector) check would give us the correct freedom space in which the user motions falls into.
2. □ If the user motions are not defined along orthogonal axes, then in order to identify a freedom space a linear independence check is required to determine the correct freedom space.

4.1.4 STEP 4: Constraint space displayed

Once the appropriate freedom space, in which the user motion falls into, is determined, the corresponding constraint space is displayed. Those corresponding constraint spaces are determined by Blanding's rule of Complimentary patterns. The constraint space is displayed as an overlay to the object. This gives the user the ability to see the constraint space with respect to the object. In the next step, the user will select specific constraints from this design space.

4.1.5 STEP 5: User selects constraints

The user now sees the constraint space overlaid on the object. The user selects $6 - n$ constraints from the constraint space where n is the number of motions defined by the user. As the user selects the virtual constraint lines, their color is changed giving visual feedback to the user. Also, the color of the selected constraint remains changed to let the user know which constraints have been selected. This visual feedback helps in error prevention from the part of the user. The user selects "Done" from the menu once he/she is satisfied from the selection. Although the lines in each constraint space are drawn as individual lines, the user understands that the constraint space consists of an infinite number of lines. In the virtual environment, we support this by drawing lines emanating from the input device in the direction and orientation of the constraint lines.

4.1.6 STEP 6: Physical constraints appear

As the user selects "Done" from the menu option, the constraint space disappears. The constraint lines selected by the user turn into physical constraints.

5. CASE STUDY: A ball joint example

This section demonstrates the methodology for the design of a spherical ball-joint example. Ball joints are useful elements in mechanism design as they only allow rotations along three axes and all the translations are restricted. In this example, the user defines rotation motions in three orthogonal directions and the system automatically comes up with the corresponding constraint spaces to choose from. It is in the final step of selection from the design space where the virtual environment is most beneficial.

5.1 STEP 1: User defines motion

As it could be difficult for a user to define three perfectly orthogonal rotation motions, we give the user presets for this

task. The system presents the user with pre-defined sets of motions which could be difficult to define or are most commonly used. Some examples could be three orthogonal rotations, two translations (x and y) and one rotation (z), two rotations (y and z) and one translation(x) etc. Along with the above presets, the user retains the ability to grab and move the object to define the motions.

5.2 STEP 2: Twist vectors calculated

The twist vectors for 3 orthogonal motions along the axis are as given below.

Table 3 : Twist representations of motion

| Motion | Twist Vector |
|-----------------|--|
| X axis rotation | $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ |
| Y axis rotation | $\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ |
| Z axis rotation | $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ |

As the movement is pure rotation, the bottom three components remain null. The three independent motions are combined to form a single resultant motion. The resultant twist vector is represented as

$$\text{Resultant Twist vector} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

5.3 STEP 3: Freedom space determined

The algorithm, instead of going through all the TYPES and CASES, just goes through the TYPES defined in CASE 3 to check for a proper match. The match is found by comparing the twist representation of the user motion with those of the freedom spaces.

Once the match is found, the search ends. For this example, the system determines that the user motion falls into CASE 3 TYPE 4.

5.4 STEP 4: Constraint space displayed

Once the exact TYPE is found, the constraint space corresponding to the freedom space would be displayed for the user to manipulate. The corresponding constraint spaces for CASE 3 TYPE 4 is shown below. The figure on the left is the constraint space of the corresponding freedom space on the right.

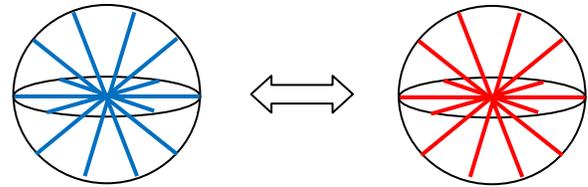


Figure 7: Constraint - Freedom space set

The constraint space is displayed as an overlay on the object. This way the user gets the idea of the location of the constraint space with respect to the object.

5.5 STEP 5: User selects constraints

Once the constraint space is displayed, the user could select individual constraints from the constraint space to apply to the object. All the constraint lines from the space are virtual and react to user’s input. The lines move as per the properties and characteristics of the constraint space. In this example, the sphere represents a grouping of an infinite number of lines which all intersect in the center. The constraint lines change color as soon the user’s wand is within certain proximity of any of the constraint lines. This lets the user know that he/she could select the highlighted line. When the user selects the line, the width and color of the line is again changed to give visual feedback to the user. When the user positions the constraint line at a desired location the line color remains changed to let the user know that this line has been selected and repositioned.

5.6 STEP 6: Physical constraints drawn

Once the user clicks on “Done” from the menu; the constraint lines selected from the space above are turned into physical constraints. This way, the object and the constraint lines are attributed with physical properties. The blue lines shown below represent the physical constraint attached to a fixed object. The red lines show the axis along which the rotation takes place.

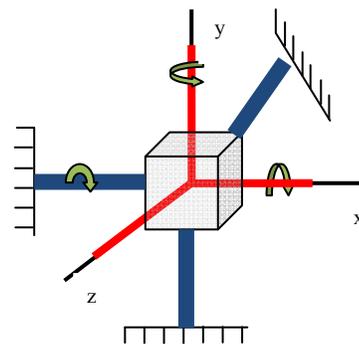


Figure 8: Final design

6. Conclusion and Future work

Currently, significant experience is required to design compliant mechanisms using the constraint based methods because of the non-intuitive motion of the compliant members. This research resulted in an intelligent design framework that will allow a broader group of engineers to design complex compliant mechanisms, giving them new options to draw upon when searching for design solutions to critical problems. The user centered strategy followed in this research is novel in the way that it frees the user from complex mathematical calculations and lets him/her concentrate on defining the desired motion and selecting from a wide range of possible solutions. The research will result in novel mechanism solutions for manufacturing and product design which have fewer movable joints, are more robust, and are easily scaled to meet the needs of micro-products.

Currently a case study is given as an example to demonstrate the proposed approach. This is a proof of concept which proves the method. The 6 step process followed gives a detailed description of how to proceed with the mechanism design process. Although the proof of concept is ready, much work still needs to be done to expand the scope. As of now, only the user motions which belong to CASE 3 TYPE 1, 4 and 5 will be recognized by the software.

Additional improvements involve support for validating the motion of the final design and further refinements of the virtual design environment.

6 Acknowledgements

This work was supported the National Science Foundation grant 0457041. The authors particularly wish to thank our collaborators at MIT: Dr. Martin Culpepper and Jon Hopkins. In addition, we are indebted to Dr. Hai-Jun Su for his efforts in this work.

7 References

- [1] □ Howell, L. L., *Compliant Mechanisms*. John Wiley & Sons, Inc., New York, 2001
- [2] □ Howell, L. L., and Midha, A., 1995. "Parametric deflection approximations for end-loaded, large deflection beams in compliant mechanisms". *ASME Journal of Mechanical Design*, 117(1), pp. 156–165.
- [3] □ Ananthasuresh GK, Kota S, Kikuchi N. Strategies for systematic synthesis of compliant MEMS. *Proceedings of the 1994 ASME Winter Annual Meeting*, November 1994.
- [4] □ Frecker, M. I., Ananthasuresh, G. K., Nishiwaki, S., Kikuchi, N., and Kota, S., 1997. "Topological synthesis of compliant mechanisms using multi-criteria optimization". *ASME Journal of Mechanical Design*, 119(2), pp. 238–245.
- [5] □ Sigmund, O., 1997. "On the design of compliant mechanisms using topology optimization". *Mechanics of Structures and Machines*, 25(4), pp. 493–524.

- [6] □ Wang, M. Y., Chen, S., Wang, X., and Mei, Y., 2005. "Design of multimaterial compliant mechanisms using level-set methods". *ASME Journal of Mechanical Design*, 127(5), pp. 941–956.
- [7] □ Hull, P. V., and Canfield, S., 2006. "Optimal synthesis of compliant mechanisms using subdivision and commercial fea". *ASME Journal of Mechanical Design*, 128(2), pp. 337–348.
- [8] □ Zhou, H., and Ting, K.-L., 2005. "Topological synthesis of compliant mechanisms using spanning tree theory". *ASME Journal of Mechanical Design*, 127(4), pp. 753–759
- [9] □ Howell, L. L., and Midha, A., 1996. "A loop-closure theory for the analysis and synthesis of compliant mechanisms". *ASME Journal of Mechanical Design*, 118(1), pp. 121–125.
- [10] □ Su, H.-J., and McCarthy, J. M., 2006. "A polynomial homotopy formulation of the inverse static analysis of planar compliant mechanisms". *ASME Journal of Mechanical Design*, 128(4), pp. 776–786.
- [11] □ Maxwell, J. C., 1890. *General considerations concerning scientific apparatus*. Dover Press.
- [12] □ Hopkins J.B., "Design of parallel flexure systems via Freedom and Constraint topologies (FACT)", PhD Thesis, MIT Cambridge, MA
- [13] □ Shai, O., and Pennock, G. R., 2006. "A study of the duality between planar kinematics and statics". *ASME Journal of Mechanical Design*, 128(3), pp. 587–598
- [14] □ Ball, R. S., 1998. *The Theory of Screws*. Cambridge University Press, Cambridge, England Originally published in 1876 and revised by the author in 1900, now reprinted with an introduction by H. Lipkin and J. Duffy).
- [15] □ Hunt, K. H., 1978. *Kinematic Geometry of Mechanisms*. Oxford University Press, New York, NY.
- [16] □ Phillips, J., 1984. *Freedom in Machinery. Volume 1, Introducing Screw Theory*. Cambridge University Press, Cambridge, U.K.
- [17] □ Phillips, J., 1990. *Freedom in Machinery. Volume 2, Screw Theory Exemplified*. Cambridge University Press, Cambridge, U.K.
- [18] □ Kong, X., and Gosselin, C. M., 2004. "Type synthesis of 3-dof translational parallel manipulators based on screw theory". *ASME Journal of Mechanical Design*, 126(1), pp. 83–92.
- [19] □ Kim, C. J., 2008. "Functional characterization of compliant building blocks utilizing eigentwists and eigenwrenches". In *Proceedings of ASME IDETC/CIE 2008*. August 3-6, New York, USA.
- [20] □ Su, H.J., Vance J.V., Dorozhkin, D.V., "A screw theory approach for the conceptual design of compliant mechanisms with flexures" Submitted to the *ASME journal of mechanism design*

From The Research Investment, Inc.

Please attach to each article delivered to your end users.

Disclaimer:

The article is for individual use only and may not be further reproduced or stored, physically or electronically without written permission from the Copyright Holder. Unauthorized reproduction may result in financial and other penalties.

Notice: Warning:

This document must not be forwarded electronically after it has been downloaded. It must not be stored in electronic format and must be deleted after a single copy has been printed.

Please contact us if there are any problems with transmission.

Phone: 216-752-0300

Fax: 216-752-0330

E-mail: orders@researchinvest.com

To all Patrons:

This article will be purged in 14 days. *If you need pages re-sent, please notify us within this time frame. Otherwise, we will have to relocate the item and you may be charged again.*

Pictures, graphs, and charts may not maintain optimal resolution after scanning. Please call 216-752-0300, or email, if hardcopy is needed.

Thank you

The Research Investment, Inc.
Phone: (216) 752-0300
Fax: (216) 752-0330
E-mail: orders@researchinvest.com