8-2009

A Virtual Reality Interface for the Design of Compliant Mechanisms

Utkarsh Seth
Iowa State University, useth@iastate.edu

Hai-Jun Su
University of Maryland, Baltimore County

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

A Virtual Reality Interface for the Design of Compliant Mechanisms

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 approach 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 is suggested where the interface guides the designer throughout the design process, thus reducing the reliance on intuitive knowledge. This methodology supports constraint-based design methods by linking mathematical models to support compliant mechanism design in an immersive virtual environment. A virtual reality (VR) immersive interface 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 identify appropriate constraints. The result is 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
VRAC, Design, Virtual reality, Compliant mechanisms

Disciplines
Computer-Aided Engineering and Design

Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2009
August 30 - September 2, 2009, San Diego, California, USA

DETC2009-87155

A VIRTUAL REALITY INTERFACE FOR THE DESIGN OF COMPLIANT MECHANISMS

Utkarsh Seth*
Department of Mechanical Engineering
Virtual Reality Application Center
Iowa State University
Ames, IA 50011 U.S.A.
useth@iastate.edu

Hai-Jun Su
Department of Mechanical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250 U.S.A.
haijun@umbc.edu

Judy M. Vance
Department of Mechanical Engineering
Virtual Reality Application Center
Iowa State University
Ames, IA 50011 U.S.A.
jmvance@iastate.edu
ASME Fellow

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 approach 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 is suggested where the interface guides the designer throughout the design process, thus reducing the reliance on intuitive knowledge. This methodology supports constraint-based design methods by linking mathematical models to support compliant mechanism design in an immersive virtual environment. A virtual reality (VR) immersive interface 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 identify appropriate constraints. The result is 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. The approach 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. The approach described in this paper helps the designer in visualizing motions and ultimately designing the desired mechanism by implementing the constraint-based approach mathematically within the user interface of a virtual reality environment.

2 BACKGROUND

Martin Culpepper and Jonathan Hopkins at the Precision Engineering Lab at MIT [12] have extended Blanding's theory [13] 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). The freedom space or freedom topology represents the object's allowable motion in space. The constraint space represents the restricted motions in space. FACT presents the different constraint and freedom spaces by dividing them into different CASEs and TYPEs. This organizes all of 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 active constraint which results in five degree-of-freedom motion. Each TYPE within a CASE defines a specific way 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 unique corresponding 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 this method can be used to design compliant mechanisms.

The key to our approach is the linking of screw theory to the geometric basis of the FACT design method. Screw theory has successfully been used in rigid body mechanism design to describe general motion of a rigid body. Geometrically, a *screw* represents a rotation about a line in space and a translation along that line. This line is known as the screw axis. Mathematically a screw motion is described with a twist vector, **T**, which is a six dimensional vector representing the linear and angular velocities of the body, written as

$$\hat{\mathrm{T}} = (\mathbf{\Omega} \;\vdots\; \mathbf{V}) = (\omega\mathbf{s} \;\vdots\; \mathbf{c}\times\omega\mathbf{s} + v\mathbf{s}) = (\omega\mathbf{s} \;\vdots\; \mathbf{c}\times\omega\mathbf{s} + p\omega\mathbf{s}) \quad (1)$$

where $\mathbf{\Omega}$ is the angular velocity, **V** is the linear velocity, **s** is the vector denoting the twist axis, **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,* **W***,* which consists of two vectors representing a force **F** and a couple (moment) **M** acting on a rigid body, written as,

$$\hat{\mathrm{W}} = (\mathbf{F} \;\vdots\; \mathbf{M}) = (f\mathbf{u} \;\vdots\; \mathbf{r}\times f\mathbf{u} + m\mathbf{u}) = (f\mathbf{u} \;\vdots\; \mathbf{r}\times f\mathbf{u} + qf\mathbf{u}) \quad (2)$$

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

These two concepts are often known as duality [14] in kinematics and statics. Screw theory has been applied to the constraint based compliant mechanism design approach. Ball [15] was the first to formulate screw theory in a systematic way. Hunt [16] and Phillips [17, 18] 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 [19]. Kim [20] studied the characterization of compliant building blocks by utilizing the concept of eigentwists and eigenwrenches based on screw theory. Researchers 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

$$\Pi_T = \begin{bmatrix} \hat{\mathrm{T}}_1 \\ \hat{\mathrm{T}}_2 \\ \vdots \\ \hat{\mathrm{T}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{\Omega}_1 & \vdots & \mathbf{V}_1 \\ \mathbf{\Omega}_2 & \vdots & \mathbf{V}_2 \\ \vdots & \vdots & \vdots \\ \mathbf{\Omega}_f & \vdots & \mathbf{V}_f \end{bmatrix} \quad (3)$$

where f is the dimension of the freedom space and $\hat{\mathrm{T}}_i$ 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

$$\hat{\mathrm{T}} = k_1\hat{\mathrm{T}}_1 + k_2\hat{\mathrm{T}}_2 = (k_1\mathbf{\Omega}_1 + k_2\mathbf{\Omega}_2 \;\vdots\; \mathbf{c}\times(k_1\mathbf{\Omega}_1 + k_2\mathbf{\Omega}_2)) \quad (4)$$

where $\hat{\mathrm{T}}_1 = (\mathbf{\Omega}_1 \;\vdots\; \mathbf{c}\times\mathbf{\Omega}_1)$ and $\hat{\mathrm{T}}_2 = (\mathbf{\Omega}_2 \;\vdots\; \mathbf{c}\times\mathbf{\Omega}_2)$ are the joint axes and the coefficients k_1 and k_2 can be viewed as the angular speeds of the joints. And **c** is the intersection point. Any motion in this space is a rotation around the axis in the direction $k_1\mathbf{\Omega}_1 + k_2\mathbf{\Omega}_2$ through the point **c**. More details concerning the relationship between screw theory and compliant mechanisms can be found in [21].

3 PROPOSED DESIGN FRAMEWORK

FACT theory, as explained, provides geometric representations of the freedom and constraint spaces. In practice, the use of this theory relies heavily on the designer's understanding of how flexible bodies move with respect to various constraint elements. Screw theory, on the other hand, gives a mathematical approach to the solution; however, it abstracts out any involvement of the designer and does not take advantage of a user-centered design approach.

While these two approaches result in successful compliant mechanism designs, specialized skills are required to use either of these approaches. Our approach combines both approaches and an immersive computer interface to facilitate compliant mechanism design. The immersive environment 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.

To begin, before the design process even starts, a catalog of the twist vector representations of all of the freedom spaces is pre-calculated and stored. When the user defines a desired motion, the twist vector representations of that motion is calculated and compared to the catalog of twist vector representations of all of the freedom spaces in order to identify the corresponding CASE and TYPE to match the desired motion. Once the freedom space is identified, the corresponding constraint space is presented to the user. The constraint space consists of a multitude of individual solution configurations from which the user picks the desired constraints that define the final mechanism.

For the proof of concept, we have developed a catalog for CASE 3 TYPEs 1, 4 and 5 freedom spaces. The geometric representations are independent of orientation and location in space. It is to be noted that the freedom space definitions are coordinate system independent. To be used in a design sense, the freedom spaces must first be aligned with a local axis system. To illustrate, we explain the twist vector representation of CASE 3 TYPE 1.

The CASE number represents the number of constraints applied on the body. For CASE 3 TYPE 1, the case number is 3 which means 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 in Figure 1 [12]. The freedom space is represented by a hoop and a plane. The hoop freedom space represents a translation along the direction perpendicular to the plane in which the hoop resides. The plane freedom space represents rotation about axes aligned with either side of the plane.

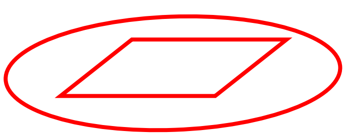
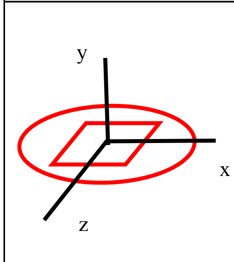
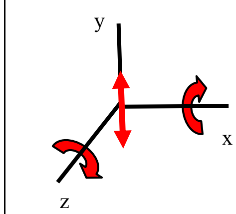
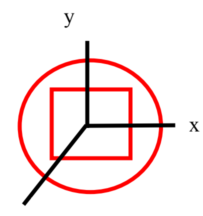
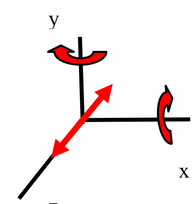
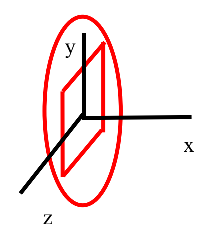
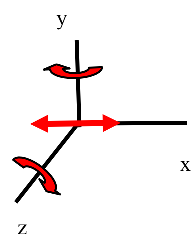


FIGURE 1: CASE 3 TYPE 1

The freedom space is oriented to the local coordinate system to match the desired input motion. This provides the context for the twist vector representation. From the definition of the twist vector, the top three elements of the vector represent rotation and last three elements represent the translation components.

TABLE 1 : CASE 3 TYPE 1
(FOR A GIVEN COORDINATE SYSTEM)

<table>
<tr><th>Freedom space in 3 orientations</th><th>Allowed motions</th><th>Twist representations</th></tr>
<tr><td></td><td></td><td>$\begin{bmatrix}1\\0\\1\\0\\1\\0\end{bmatrix}$</td></tr>
<tr><td></td><td></td><td>$\begin{bmatrix}1\\1\\0\\0\\0\\1\end{bmatrix}$</td></tr>
<tr><td></td><td></td><td>$\begin{bmatrix}0\\1\\1\\1\\0\\0\end{bmatrix}$</td></tr>
</table>

4 PROPOSED SOLUTION

We propose an approach which uses the above approach for the design of compliant mechanisms. The solution proposed below in section 4.1 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 Detail design steps

A user begins 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. The 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 accomplished through the use of a menu selection. 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 calculates the possible constraint spaces, which, when applied to the rigid body, allow the defined motion path. The user can select appropriate constraints from the space (guided by design principles) which then result in a final mechanism design. This approach is outlined in Figure 2.

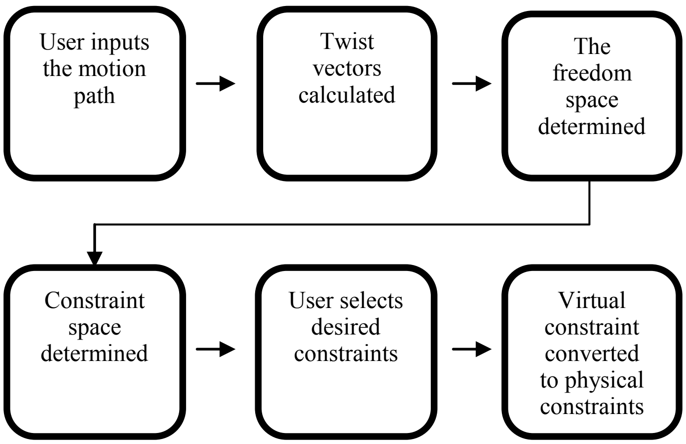


FIGURE 2: DESIGN METHODOLOGY

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 (Fig. 3). 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. This is done to assist the user as he/she might find it difficult to locate it back to the exact original position to define the next motion. Once the user is finished, the system has n + 1 number of matrices ('n' number of positions & 1 starting position where $n \leq 6$).

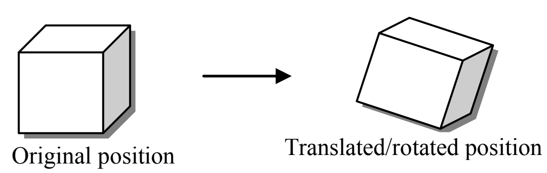


FIGURE 3: OBJECT TRANSFORMED TO A NEW LOCATION

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. The number of twist representations depends upon the number of motions defined by the user. They are then combined to form into a one single twist vector which represents the whole user motion.

4.1.3 STEP 3: Freedom space determined

After the completion of the second step, the system has the twist representation of all the 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. 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 two 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.
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 is determined, the corresponding constraint space is displayed. Those corresponding constraint spaces are determined by Blanding's rule of complimentary patterns [13]. 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 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 are redisplayed as elements of the compliant mechanism.

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 in Table 3.

TABLE 2 : 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 in Figure 4. The figure on the left is the constraint space of the corresponding freedom space on the right.

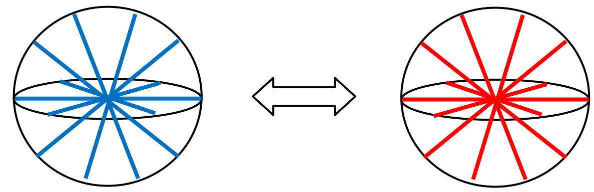


FIGURE 4: CONSTRAINT SPACE (LEFT) AND FREEDOM SPACE (RIGHT) SET

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. The sphere geometry of CASE 3 TYPE 4 represents an infinite number of possible constraint lines all intersecting at a single point. The lines are selectable by the user and conform to the restrictions of the constraint space. 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 constraint space define the compliant mechanism. The constraints represent idealized links with compliance only in the direction perpendicular to the axis of the element. The blue lines shown below in Figure 5 represent the compliant links attached to a fixed object. The red lines show the axis along which the rotation takes place.

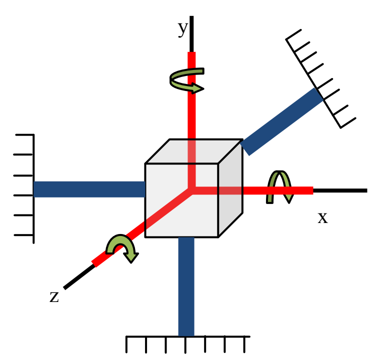


FIGURE 5: FINAL DESIGN

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 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.

7 ACKNOWLEDGEMENTS

This work was supported the National Science Foundation grant CMMI-0457041. The authors particularly wish to thank our collaborators at MIT: Dr. Martin Culpepper and Jon Hopkins.

8 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] Blanding, D.L., 1999. *Exact Constraint: Machine Design Using Kinematic Processing*. ASME Press, New York, NY.

[14] 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.

[15] 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).

[16] Hunt, K. H., 1978. *Kinematic Geometry of Mechanisms*. Oxford University Press, New York, NY.

[17] Phillips, J., 1984. *Freedom in Machinery*. Volume 1, Introducing Screw Theory. Cambridge University Press, Cambridge, U.K.

[18] Phillips, J., 1990. *Freedom in Machinery*. Volume 2, Screw Theory Exemplified. Cambridge University Press, Cambridge, U.K.

[19] 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.

[20] Kim, C. J., 2008. "Functional characterization of compliant building blocks utilizing eigentwists and eigenwrenches". *Proceedings of ASME IDETC/CIE 2008*. August 3-6, New York, USA.

[21] Su, H., Vance, J. M., Dorozhkin, D. V., 2009. "A screw theory approach for the conceptual design of compliant mechanisms". *Proceedings of the ASME International Design Technical Conferences and Computers in Engineering*, Aug. 30 - Sept 2, San Diego, CA, DETC2009-86684.

